# REQUIREMENTS SPECIFICATION PART I
# and
# ANALYSIS OF DYNAMIC SIMULATION METHODS
# for
# LAUNCH VEHICLE COMPONENT LEVEL SIMULATION

MARCH 1966

PREPARED UNDER MODIFICATION NO. 1 TO
CONTRACT NAS8-20060, DCN 1-5-60-00036-01 BY

APOLLO SUPPORT DEPARTMENT
MISSILE AND SPACE DIVISION
GENERAL ELECTRIC
DAYTONA BEACH, FLORIDA

MARSHALL SPACE FLIGHT CENTER
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

PRECEDING PAGE BLANK NOT FILMED.

## TABLE OF CONTENTS

TABLE OF CONTENTS (CONT.)

TABLE OF CONTENTS (CONT.)

## LIST OF ILLUSTRATIONS

## LIST OF ILLUSTRATIONS (Cont.)



## LIST OF TABLES

# SECTION 1

# INTRODUCTION

An analytical study of Launch Vehicle Component Level Simulation (LVCLS) was conducted from March 1965 through November 1965. The report on this previous study effort was issued in December 1965. The previous study showed that launch vehicle simulation at the component level is practical and desirable. The current study extension was undertaken to develop a requirements specification for those outputs being considered by MSFC for early implementation and to compare, by computational means, several dynamic simulation methods to select the method or combination of methods best suited to the solution of the problems encountered during a launch vehicle system simulation.

## SECTION 2

## STUDY OBJECTIVES

There were two basic study objectives. The first objective was to generate a requirements specification Part I for the CPCEI "Launch Vehicle Component Level Simulation Computer Program" which provides outputs A, B, C, K, L, M, N, O, P, Q, R, $S_1$, $S_2$, $S_3$, U, Y, Z, AA, AB, AC, AF, AG, AH, AI, and AJ for uses II, III, V, VI, VII, VIII, XIII, XV, XVII, and XIX as defined in modification No. 1 to contract NAS 8-20060 and the December 1965 report on the Analytical Study of LVCLS. The second objective was to compare dynamic simulation methods by computational means to select the best method or combination of methods for use by LVCLS. Results of these study efforts are reported herein.

## SECTION 3

## SUMMARY OF RESULTS AND RECOMMENDATIONS

### 3.1 GENERAL

The LVCLS computer program described in the requirements specification is intended to be exceptionally flexible. Standard computational modules will be related to each launch vehicle or GSE system for utilization in an automatic mode of operation. Where the user requires more accuracy, less running time or otherwise wishes to select a different available computational technique for use on a particular problem, he may directly input his requirements and override most normal combinations. The capabilities and limitations of each computational technique will be explained in a user's manual so that the engineer will be aware of the significance of the changes he requests. As discussed under the dynamic simulation approaches, no one technique is best for all problems nor for all criteria on any one problem. Each vehicle system will be identified with a normal technique and with applicable alternate modes for manual selection.

### 3.2 REQUIREMENTS SPECIFICATION

The LVCLS computer program contract end-item requirements specification Part I is given in Section 4. LVCLS is not a computer-based system with an accurately predefined set of problems to solve. Rather, it is an engineering tool for use on computational laboratory computers where an individual engineer selects the hardware systems and conditions for simulation of problems specified at the time of use. Consequently, the engineering data base and computer programs which make up the simulation system must be developed together to provide the required flexibility. Each set of equipment will have a normal set of outputs provided from a simulation run. The using engineer may choose from those available outputs and generate an output report in his own format if desired. As an example, normal output from the gyro table X-loop might be:

- Seven loop currents versus time
- Torquer input voltage versus time
- Torquer output voltage or torque versus time
- Angle $\beta$ versus time
- Velocity or rate of change of angle $\beta$ versus time
- Acceleration or second derivative of angle $\beta$ versus time

- Input voltage versus time
- Power consumption versus time.

These quantities for the most part are calculated in the process of solving the system dynamics and are available from a computational run. The report generated for the user will edit these and provide only the data required. Should this user or another desire other data for the same problem conditions at a later time, it will not be necessary to recompute the data. Instead, the original output tape may be edited to provide the additional output.

Should the using engineer chose to override the normal computational method which provided the above detailed output, he could select a computational mode which evaluates a single overall function. In this case his only available outputs would be:

- Angle $\beta$ versus time
- Input voltage versus time.

The computation could be done somewhat faster using a single overall equation, but internal currents, voltages, etc. would not be provided, nor would they be recoverable.

## 3.3 DYNAMIC SIMULATION APPROACHES

In the earlier study, several methods of solving the dynamic equations of the Launch Vehicle Component Level Simulation were proposed. These were:

a. Direct solution by numerical integration of differential equations
b. Analytical determination of system transfer function from subsystem or component transfer matrix
c. Numerical determination of system transfer function by root finding and curve fitting
d. Calculation of system time response by number series from subsystem or component time responses.

In order to compare these methods, they have been tried on a representative portion of the Inertial Measurement Unit, the X-loop of the ST-124-M Inertial Platform. The results showed that no one method stands out as being superior in all cases. A proper combination of methods for different portions of the system will give optimum results.

# SECTION 4

# REQUIREMENTS SPECIFICATION

# PART I

FORM A

Specification No. __________
Revision No. __________
Release Date __________

CONTRACT END ITEM DETAIL SPECIFICATION
COMPUTER PROGRAM

PERFORMANCE/DESIGN
AND
PRODUCT CONFIGURATION
REQUIREMENTS

CEI ###

LAUNCH VEHICLE COMPONENT LEVEL SIMULATION COMPUTER PROGRAM
FOR
SATURN LAUNCH VEHICLE AND G.S.E.

APPROVED BY: __________

DATE: __________

CONTRACT NO.: __________

APPROVED BY: __________

APPROVAL DATE: __________

FORM B

Specification No. ________
Revision No. 0
Release Date ________

CONTRACT END ITEM DETAIL SPECIFICATION
COMPUTER PROGRAM

PART I

PERFORMANCE/DESIGN
REQUIREMENTS

CEI ###

LAUNCH VEHICLE COMPONENT LEVEL SIMULATION COMPUTER PROGRAM
FOR
SATURN LAUNCH VEHICLE AND G.S.E.

APPROVED BY: ________

DATE: ________

CONTRACT NO.: NAS 8-20060

APPROVED BY: ________

APPROVAL DATE: ________

## 1.0 SCOPE

This part of this specification establishes the requirements for performance, design, test, and qualification of one type-model-series of equipment identified as Launch Vehicle Component Level Simulation, CPCEI ###. This CPCEI will provide a complete, integrated, and dynamic computer simulation of the launch vehicle and its support equipment that will aid in:

- Equipment design
- Verification of equipment functional relationship
- Reliability and safety evaluations of equipment and procedures
- Test and evaluation of the functional operation of equipment
- Development of checkout operations and procedures
- Validation of manufacturing tests
- Configuration management, identification, accounting, and control
- Growth and development of advanced systems.

The CPCEI requires the generation of a data base which describes functionally the vehicle configuration and the development of the logic or programs required to operate on these data. Although this CPCEI is primarily directed to simulation of the Saturn V launch vehicle and the GSE, the logic or programs must be capable of processing other configurations given a properly structured data base. The simulation must recognize events of both a discrete and dynamic nature. The data base must be structured such that the complete hardware system or a selected portion can be simulated. The type of output required and the problem identification must be under the control of potential users of varying skills and disciplines but sufficiently cognizant of the simulation capabilities and limitations. To this end this CPCEI shall be designed to accept user-oriented language and provide operating modes ranging from automatic or semi-automatic to manual select. A manual describing input requirements, program, and subprogram operations for each of the operating modes, and error and legality checks must be prepared.

## 2.0 APPLICABLE DOCUMENTS

The following documents, of exact issue shown, form a part of this specification to the extent specified herein. In the event of conflict between documents referenced here and the other detailed content of Sections 3 and 4, the detailed requirements of Sections 3 and 4 shall be considered as a superseding requirement.

## 2.1 PROJECT DOCUMENTS

### 2.1.1 SPECIFICATIONS

Configuration Management of Computer Program Contract End Items (A Proposed Exhibit to NPC 500-1), 11 June 1965.

## 2.2 OTHER PUBLICATIONS

The MSFC Configuration Management Accounting System, R-COMP-A-66-1, 5 January 1966.

Analytical Study of Launch Vehicle Component Level Simulation, prepared under contract NAS 8-20060, DCN 1-5-60-0036-01 by Apollo Support Department, Missile and Space Division, General Electric Company, Daytona Beach, Florida, dated December 1965.

Launch Vehicle Component Level Simulation Supplemental Scope of Work, Appendix "A," Modification No. 1, Contract NAS 8-20060, dated 26 November 1965.

## 3.0 REQUIREMENTS

In the case of simulation of existing and/or proposed physical systems, test and check-out of the computer program ultimately hinge on the experience and judgment of the simulation designers and, the hardware system designers in determining that the model, as reflected by the program, does in fact reasonably represent the physical system. In order to help insure that the final program product will fit the above requirement, the following procedure should be followed:

- Scope out the system to be simulated in terms of the essential hardware elements that constitute the system, the physical connections between elements, the boundary conditions that exist at all pertinent interfaces with external systems, and the expected steady state system conditions at all design points.
- Write component equations of the identified elements in terms of the physical characteristics of the element and its boundary conditions. It is essential that the set of equations which physically describe the elements satisfies the requirements of conservation of energy, mass, and momentum, and that the set meets the approval of the system and component designers.
- Once the system equations are determined and agreed upon, it is necessary to begin the mathematical analysis (of these equations) required to specify the computer program. The mathematical analysis is required to design a computer program which is computationally stable and accurate and which, in general, avoids the need to compute around algebraic loops.

  Numerical stability generally will be evident in terms of the transient deviation of the computer solution from the true solution.

  It should be recognized that in a complex system of equations it is generally not possible to get a direct measure of the deviations due to the computer itself. The only satisfactory approach is (1) Design the computer devices (in the case of analog) or numerical methods (in the case of digital) to minimize the introduction of computer phase shift; (2) Test sample problems which are representative of the degree of nonlinearity and coupling but amenable to alternate methods of checking; and (3) Program the set of system equations and introduce provisions for running parts of it under transient and steady state conditions.

  In the ultimate program, the only thing that can be verified with any degree of accuracy is the steady state conditions. The overall approach outlined above simply enhances ones level of confidence in the computer performance.

Once the program appears acceptable, a set of steady state and transient runs should be made for future checks against variations that will be made in the program.

The problem of algebraic loops must be handled in the initial mathematical analysis. These loops must be recognized and broken. Failure to do so will result in marginal computer performance and (in the digital computer) will result in excessive running time.

The above ideas present general guidelines for the development of a realistic simulation program. It is not feasible to set down hard and fast rules since each system is individual from the standpoint of physical characteristics and of degree of sophistication required in the modeling. It should be pointed out, also, that application of the guidelines is a matter of degree. The degree to which the rules are applied will depend on three basic factors:

- Experience and judgement
- Time
- Cost.

Experience and judgement generally result in more rapid test and checkout without sacrificing quality. The constraints of time and cost generally are dominant elements. All simulation programs must be developed within a fixed amount of time and at a fixed cost. This means of course that the simulation designer must carefully estimate the degree of sophistication in the model and the degree of program testing that will be required to bring the model within an agreed upon level of performance.

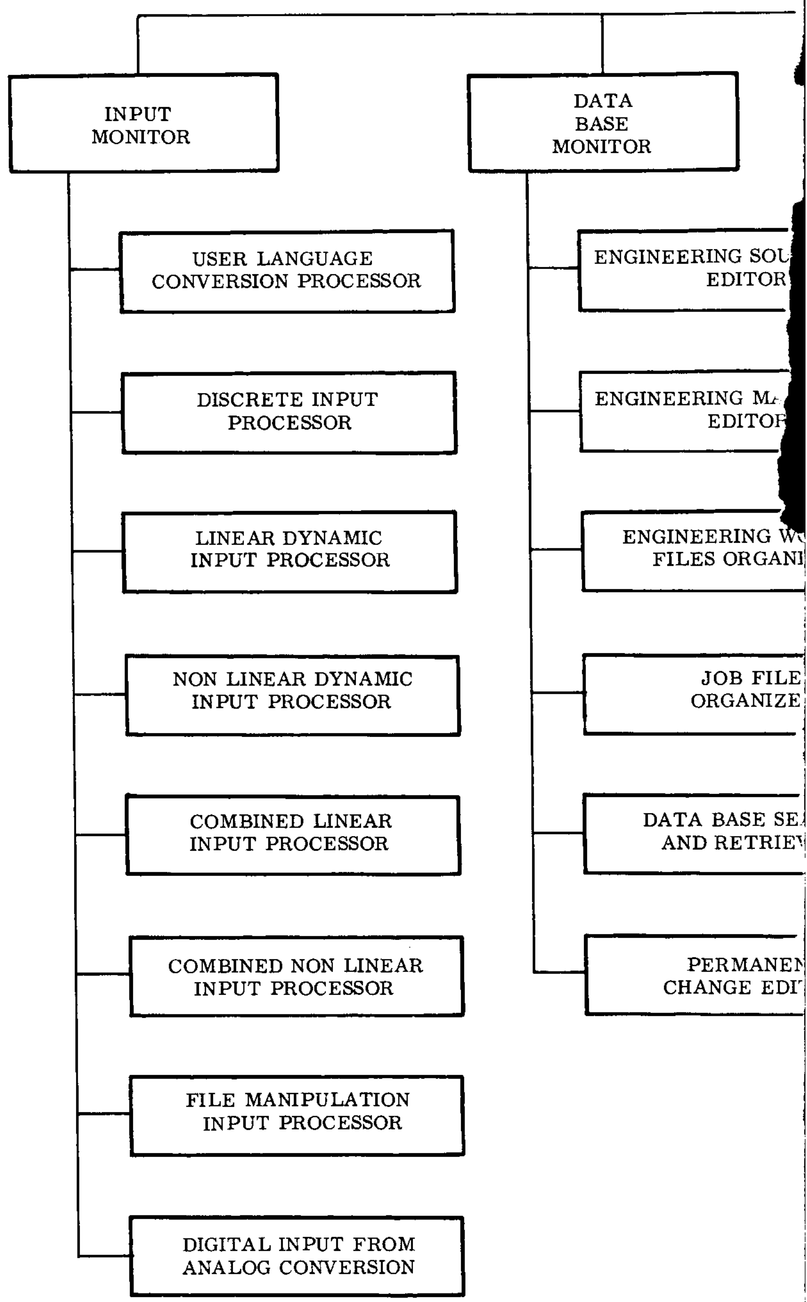
Final verification of the adequacy of the model depends not only on the user's demand for output, but also on how well it fits the real system. The prime question here is either that of how to justify using the model at hand or to justify refining it. Obviously, it is assumed at this point that simulation models generally do not exactly match the characteristics of the real system. It is essential that the simulation and real system be subjected to the same input and that the outputs be compared. Analysis of the output data will lead to a measure of the accuracy of the simulation. Whether or not the model is considered acceptable will have to be resolved within the framework of the nature of the system, the application of the simulation, the degree of importance of the system in overall system operations, and the degree of difficulty, time, and cost that must be expended in improving overall model accuracy.

## 3.1 PERFORMANCE

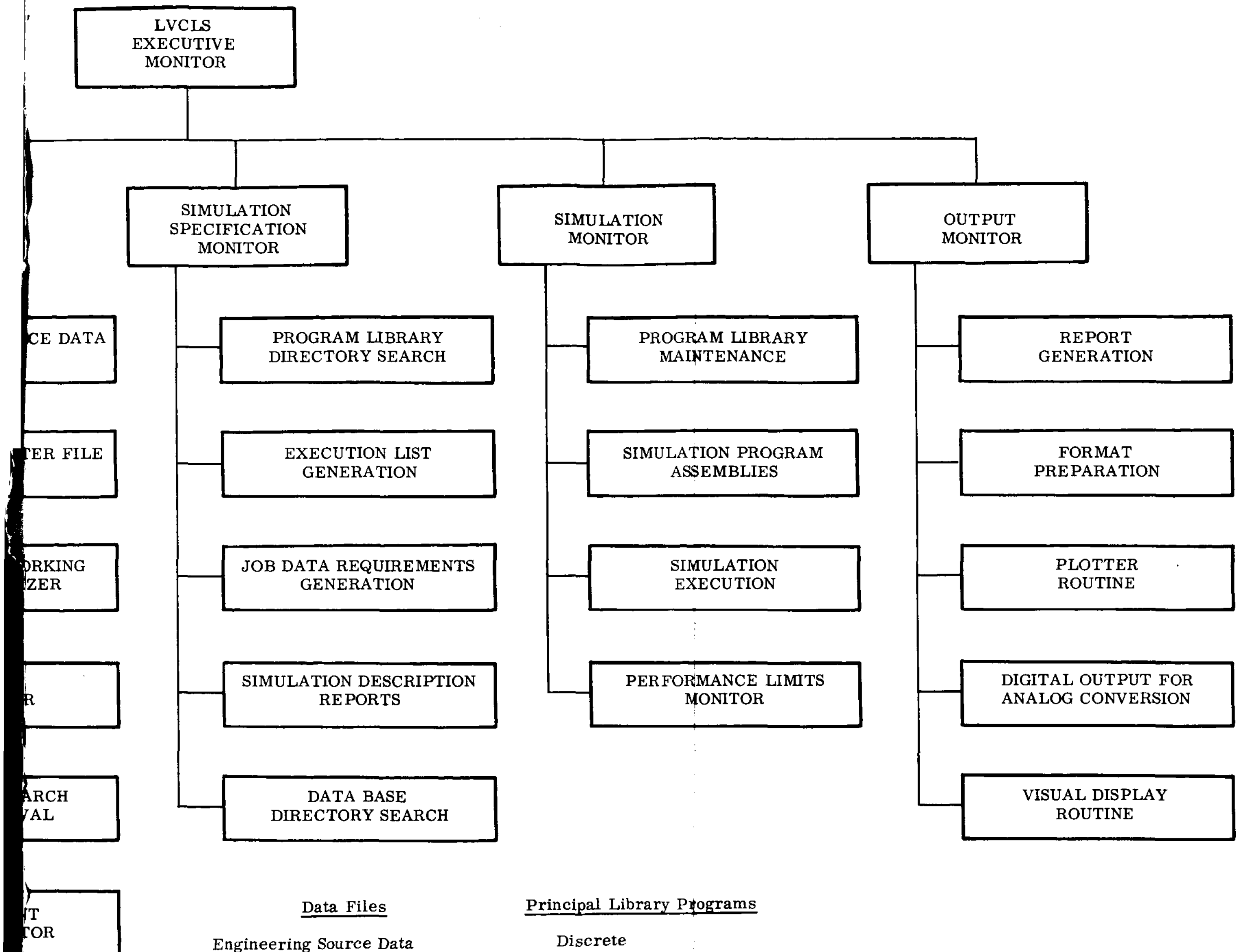
The computer simulation system described herein is composed of many related programs including data preparation organized under an Executive Monitor and simulation submonitors. This simulation system is capable of preparing a functional data base describing desired vehicle systems from its engineering source data file and performing discrete and/or dynamic computer simulations requested to produce required engineering outputs. The entire simulation is performed under control of standard IBSYS and IBJOB monitors or under their equivalent supplied with computer hardware. Each program module is compiled prior to insertion into the computer program library. After the initial compilations the computer simulation system operates in machine language for organizing a particular job data file and organizing the job program structure.

Internal control of the simulation system is exercised by its own Executive Monitor. In a typical computer run, the Executive Monitor reads input which specifies the particular type of computational job to be performed, and then passes control to the Input Monitor which selects the proper Input Processor for the job. The selected Input Processor prepares required input functional data using its source data file and incorporating changes requested by the user's input. When input data have been prepared, control is returned to the Executive Monitor which then selects the proper Job Monitor to organize the required simulation programs and functional data. The Job Monitor executes a simulation and returns control to the Executive Monitor which then passes control to the Output Monitor. The Output Monitor edits simulation output, organizes it into proper format, lists, plots, or otherwise provides the required engineering outputs.

The basic program structure block diagram is given in Figure 4-1. This modular structure provides the ability to perform simulation with sets of operable programs using a partial (subsystem) data base without penalty from incomplete programs or data which do not directly relate to the immediate problem. New programs may be added to the simulation system, and improvements may be made on existing program modules with a bare minimum of changes elsewhere. Paragraph 3.1.1.2 "Processing" shows the top level functional flow for the computer simulation system with most programs in use. In normal use, the Job Monitor organizes a smaller number of programs into a more efficient operating tool.


INPUT MONITOR
DATA BASE MONITOR
USER LANGUAGE CONVERSION PROCESSOR
DISCRETE INPUT PROCESSOR
LINEAR DYNAMIC INPUT PROCESSOR
NON LINEAR DYNAMIC INPUT PROCESSOR
COMBINED LINEAR INPUT PROCESSOR
COMBINED NON LINEAR INPUT PROCESSOR
FILE MANIPULATION INPUT PROCESSOR
DIGITAL INPUT FROM ANALOG CONVERSION
ENGINEERING SOU
EDITOR
ENGINEERING M
EDITOR
ENGINEERING W
FILES ORGANI
JOB FILE
ORGANIZE
DATA BASE SE
AND RETRIE
PERMANEN
CHANGE EDI

2




<u>Data Files</u>

Engineering Source Data
(Random Disk single listing)

Engineering Master File
(Organized Disk single listing)

Engineering Working File
(Organized Disk multiple listing by configuration, subsystem, type simulation)

Job File
(Organized Core and Disk, problem limited, multiple listing, temporary)

<u>Principal Library Programs</u>

Discrete

Linear Dynamic

Nonlinear Dynamic

Combined Linear

Combined Nonlinear

Discrete Comparison

Dynamic Comparison

Figure 4-1. Executive Monitor Structure for LVCLS System

FOLDOUT FRAME

### 3.1.1 FUNCTIONAL REQUIREMENTS

The prime purpose of the LVCLS is to simulate the operation of the launch vehicle or some specified portion of it under a set of given conditions. As a result, the simulation breaks down into two distinct parts:

- The data base which describes functionally the vehicle configuration
- The logic or programs required to operate on these data.

The general features which will be provided are described in the following paragraphs; the general limitations at the end of 3.1.1. Additional details will be contained in 3.1.1.1 through 3.1.1.4.

<u>FEATURES</u>

1. Although the LVCLS is primarily directed to simulation of the Saturn V launch vehicle, the logic or programs must be capable of processing other configurations, provided that these configurations are described by a properly structured data base. Succeeding paragraphs will describe this data base structure in more detail.
2. To represent operation of the launch vehicle, the simulation must recognize events of both a discrete and a dynamic nature. For example, helium does not flow through the helium control solenoid valve in the J-2 engine system until after receipt of an electrical signal from the sequencer controller. Thus, the operation of this valve and its effect on helium flow depends on receipt of a discrete signal. The flow of helium through this valve, as a function of time, will then result from the application of the dynamic portion of LVCLS and will continue until another change of state occurs. More on this point will appear in 3.1.1.2.
3. Application of LVCLS requires that the launch vehicle data base structure must permit simulation of only a portion of the vehicle as well as of the whole thing. As an example, consider the oxidizer pressurization system for the S-IVB. Further, consider a problem in which the user desires the requirements of the system to maintain pressure in the oxidizer tank under the following conditions:
   - A given depletion rate of oxidizers into the propellant systems
   - A given flow rate and temperature for the oxidizer turbine exhaust.

   Under these conditions, the major interfaces with the propellant system have been specified. Thus, it would be possible to split off the oxidizer pressurization system and study it by itself. Without such a split, it would be necessary to consider the whole S-IVB and its associated equipment, and this alternative would result in increased processing time and cost because of the increased complexity of the latter system.

4. To better meet the requirements of the potential user, the type of output information desired must be under his control but within restrictions indicated in Analytical Study of Launch Vehicle Component Level Simulation, December 1965. The reason for this feature is evident when one considers, for example, the two types of problems below:

   - What is the sequence in which equipment operations occur?
   - What is the system response to a given signal or set of signals?

   In the first case, the user is interested only in the timing of events; in the second, he requires more detailed output. Formatting of the output will be a function of the input-output portion of LVCLS.

5. The LVCLS will consider two types of data changes, permanent and temporary. Further, record of the permanent changes will be kept for recall. The reasons for the two types of change information are as follows:

   - The permanent data base must reflect the current configuration. These changes will not appear in the data until they have received the necessary approvals as spelled out by the configuration management policies. Even then, such changes will be inserted only by selected personnel.
   - Temporary changes will not be inserted into the data base. They are the means of testing the effect of a proposed change on operation of the system. Insertion of such changes into the data base would result in possible loss of the whole base since it would no longer reflect the current configuration.

   Three main classifications of changes must be handled. They are:

   - Insertion of data describing functional components which have been added, ... e.g., relays and contacts, valves, etc., with signal and/or effluent routing added.
   - Deletion of data describing functional components which have been removed, . . . e.g., relays and contacts, valves, etc., with signal and/or effluent routing deleted.
   - Insertion or deletion of data describing changes in functional components, . . . e.g., change of signal routing to and from relay coils and contacts, change of dynamic parameters, etc.

Such changes will be communicated to the simulation in specified format.

Temporary Change Data will be used for more than one purpose. It will accomplish the following:

- Describe, in the proper format, the initial conditions, e.g., states of relays, states of contacts, valve positions, pressures, etc.
- Describe, in the proper format, states of equipments at given times. This serves to communicate to the LVCLS states of components which are the result of operator actions.
- Describe, in the proper format, equipment states that will not change during the time period under investigation. This serves to define bounds to the systems portion to be studied, e.g., a relay in de-energized state, a valve in the closed position, etc.

For checkout or countdown verification, each sequence of steps would require Temporary Change information such as that described above. In addition, such information must be outputted to identify the conditions under which the analysis was run (output AK, Table 4-1).

6. The systems making up the launch vehicle, in most cases, are highly nonlinear, e.g., the propellant systems. Thus, the differential equations describing the dynamics of such systems will be nonlinear. The dynamics of other portions of the launch vehicle may be described by linear differential equations. Thus, the simulation must be capable of processing both linear and nonlinear differential equations for prediction of system dynamics.

## LIMITATIONS

Along with the general features described above are limitations to LVCLS performance. They appear below:

1. The simulation will not perform in real time where real time is defined as the time required for the launch vehicle equipment under study to perform its function. As an example, the ratio of processing time on an IBM 7044 to the response time for the X-loop of the gyro system to a pulse is approximately 1800:1 when using an error criterion of $10^{-7}$. This does not say that the variable value versus time will not be listed. Rather, it indicates the time required to process the differential equations to obtain the response versus time.

Table 4-1

Output Codes Versus Output Description

| Outputs | | |
| --- | --- | --- |
| Code | Description | Form of Output |
| A | Function Sequence Chart | Plot |
| B | Listing of Sequence of Operations by Time | List |
| C | Listing of Component Status Change | List |
| K | Listing for Comparison Runs | List |
| L | Listing of Delay Times for Selected System Position | List |
| M | Listing for Comparison of Delay Times | List |
| N | Listing of Equipments Unstable in Operation | List |
| O | Plot of Transient Response | Plot |
| P | Listing of Equipment by Panel | List |
| Q | Listing of Equipment by Drawing Number | List |
| R | Listing of Equipment by Function | List |
| $S_1$ | Schematic of Equipment by Panel | Schematic Plot |
| $S_2$ | Schematic of Equipment by Drawing Number | Schematic Plot |
| $S_3$ | Schematic of Equipment by Function | Schematic Plot |
| U | Listing of Equipments Activated with Time or Number of Activations | List |
| Y | Configuration Accounting Report "Master Reference List—In CEI/ECP Sequence—Page 1" | Report |
| Z | Configuration Accounting Report "Master Reference List—In CEI/ECP Sequence—Page 2" | Report |
| AB | Configuration Status Accounting Report "Modification Status" | Report |
| AC | Configuration Status Accounting Report "Spares Status" | Report |
| AF | Listing of Transient Response | List |
| AG | Listing of Changes Which Affect Simulation Operation | List |
| AH | Listing of New Approved Permanent Data Being Entered | List |
| AI | Listing of New Approved Permanent Changes Being Entered | List |
| AJ | Listing of Equipments Involved Within Specified Bounds | List |

2. The results will simulate the operation of the vehicle configuration contained in the data base and the temporary changes. Any discrepancy between the data and the actual configuration will show up as a deviation between actual and simulated operation. This includes the effects of assumptions which may be made in the dynamic or logical expression.
3. The function of all simulations is simulating the process under consideration. In the case of LVCLS, this will be its only function. It will show up as output which describes the operation under given conditions of the launch vehicle configuration as indicated by the data base and temporary changes. As such, it will provide data only. For more advanced studies, e.g., improved test procedures, the user will exercise ingenuity in organizing his experiments. The analysis of them will be performed on an off-line basis, possibly at his desk or possibly through the use of subsidiary programs which are not part of LVCLS.
4. Cases will exist in which response of one signal is wanted for a set of initial conditions. This signal will always be part of a launch vehicle subsystem, and the LVCLS will imbed it into a set of signals and process this set. Editing of the desired signal will be performed by the output function.

#### 3.1.1.1 Inputs

The data required for the Launch Vehicle Component Level Simulation can be classified as Permanent Data, Permanent Change Data, and Temporary Change Data.

The Permanent and Permanent Change Data will be obtained from the Engineering Source Data prepared by MSFC. The Engineering Source Data file is part of the overall Configuration Management file which contains many data elements such as cost and schedule information which are not required for simulation purposes. The data elements listed under the headings of Permanent Data and Permanent Change Data are those which must be present in the Engineering Source Data file to support the component level simulation.

These data needs are based on the outputs noted in the report, Analytical Study of Component Level Simulation, dated December 1965. Some of the outputs in the referenced report are for the purpose of supporting the MSFC Configuration Management Accounting system. The input data elements which are of use only for these outputs are identified by the symbol (*). Another series of outputs mentioned in this report is that which provides panel, drawing, and function schematics. This output requires connection statements as input. A recent study performed for the Mechanized Graphics effort at General Electric ASD indicates that a computer program can be written which will translate connection statements into logical statements in approximately 70 percent of the cases. The remaining logical statements required for the simulation will have to be generated off-line by hand. The additional effort required to produce, store, and maintain the connection statements for the single purpose of providing this output is deemed inadvisable.

It is often useful to examine the effect of a proposed change via the component level simulation to aid in the change evaluation. The computer program shall be capable of introducing the desired changes into the data base for a particular simulation. Certain other data elements such as the initial condition and effective configuration date are also classified as temporary data.

These data shall be introduced into the Job Data File via the Input Monitor such that the permanent data files are undisturbed. A list of these data elements is provided under the heading of Temporary Change Data.

The structure of the Engineering Source Data file and that of the Temporary Change Data will be strongly influenced by the requirements of the simulation. The detailed structure and formatting of this data will be developed along with the individual computer program modules; however, some general requirements with respect to the data base can be stated. These are:

- The data base must be function oriented.
- The hardware and parameters applicable to each function must be identified.
- The logic statements, which are largely function and parameter oriented, must be related to the hardware.
- The dynamic equations must be related to the functions, discretes, and the hardware involved.
- Changes in the hardware and/or parameters must be reflected in the logic statements and the dynamic equations.
- The data base must be capable of supporting several levels of simulation to various degrees of sophistication.
- The formatting of the data must be kept as simple as possible and compatible with user requirements.

## PERMANENT DATA

Location
- Vehicle Position
- Panel
- Geographic

Nomenclature
- Component Name
- Drawing and Page Number
- Specification Number
- Specification Custodian*
- Contractor Identification*
- Contractor Number*
- Serial Number
- Specification Prepared by*
- Part Number
- Specification Schedules Issue Date*
- Configuration Control Board Number*
- Project
- Cognizant R and DO Laboratory*
- Vehicle Designation
- Contract End-Item Number
- Contract End-Item Nomenclature

Quantity*

Spares Status*

Connection Statements

Logical Statements

Dynamic Equations

Element Parameters as a function of Environment

Criteria for Equipment Operations

Expected Time Delays

Failure Rates

(*Configuration Management Accounting System only.)

## PERMANENT CHANGE DATA

Changes in any of the Permanent Data entries and in addition:
- Engineering Change Proposal Number
- Affected Part Number
- Quantity Affected*

Contractor Identification*
Number of Kits to be Procured by Contractor*
Quantity in Production to be Modified*
Quantity Requiring Modification by Contractor*
Quantity Requiring Modification by Contractor Completed*
Quantity Requiring Modification by MSFC*
Quantity Requiring Modification by MSFC Completed*
Date of Kit First Schedule*
Date of Kit Accepted*
Date of Kit Delivered*
Contract End-Item Locations
New Part Number
Type Identification*
Level of Modification*
Kit Identification*
Required Delivery Date*
Actual Delivery Date*
Expended Code*
Change Title
Contract Change Proposal Issue Date
Contract Change Notification Number
Specification Change Notice Number
Contract Change Notification*
Related Document Numbers*
Identification of Critical Components*

(*Configuration Management Accounting System only.)

## TEMPORARY CHANGE DATA

Temporary changes in data contained in the permanent data files:

Effective Configuration Date and Vehicle Designation
Output Scale—Abscissa and Ordinate
Assumed Sequence of Operation
Outputs Desired
Equipments Assumed to be Activated During Run
Functional Limits
Required Indicator Pattern

Initial Condition such as:

- Temperature
- Position
- Acceleration
- Pressure
- Flow Rate
- Current
- Voltage
- Impedance
- Event
- Time Delays
- Failure Rates
- Time

## EXAMPLES OF NECESSARY DATA

| | |
|---|---|
| Valves | Cross-sectional area as a function of pressure or voltage and time, turbulence as a function of flow and area, specific heat and mass of valve and heat sink, volume as a function of valve position, and heat conductivity |
| Pipe Lines | Cross-sectional area and length, specific heat and mass of line and heat sink, heat conductivity, and frictional factor |
| Tanks | Volume, specific heat and mass of tank and heat sink (baffle area and placement), heat conductivity, and coefficient of discharge |
| Pressure Switches | Activiation and deactivation pressures |
| Temperature Switch | Activation and deactivation temperatures |
| Timers and Controllers | Fixed delay and delay as a function of temperature, activation conditions |
| Electrical Wiring | Reactive effects on electrical signals |
| Gas Generators | Proper oxygen-fuel mixture for ignition and combustion, chamber volume, orifice coefficients of discharge and area, specific heat and mass of generator and heat sink, and efficiency of combustion |

| | |
|---|---|
| Turbines | Moment of inertia or mass and radius of gyration, efficiencies of operation, specific heat, heat conductivity, mass of turbine and heat sink, and volumes |
| Gas and Liquid | Viscosity, thermal conductivity, specific gravity, specific heat, coefficient of volume expansion with temperature, heats of vaporization, heat of combustion, equation of state in terms of energy, temperature, pressure, and specific volume or density |
| Electrical Wiring | Logic equations describing the system connections |
| Power Supplies | Output voltage or current as a function of load and input voltage, power consumption as a function of load, and frequency regulation |
| Relay Assemblies | Pickup and dropout delay times as a function of activation voltage, type of relay (locking, etc.), and power consumption |
| Modulators and Demodulators | Power consumption, output levels versus input levels |
| Telemetry | Power consumption, output as a function of input |
| Amplifiers and Preamplifiers | Power consumption, voltage or current output as a function of input power |
| Gyros | Power consumption, angular velocity of the rotating wheel, precession rate, moment of inertia of the rotating wheel, degrees of freedom, pickoff voltages versus angular displacement, and angular displacement versus input torque |
| Accelerometers | Power consumption, encoder output as a function of acceleration and input power |
| Gimbals | Moment of inertia, damping coefficient, and frictional forces |
| Resolvers, Synchros and Microsynchros | Output voltages as a function of input voltage and position angle |

| | |
|---|---|
| Motors | Angular velocity as a function of input voltage |
| Torquers | Angular displacement as a function of input voltage |
| Sensors | Timers, heaters, switches, blowers, and ducts |
| Chambers | Volume, specific heat and mass of chamber and heat sink, heat conductivity, proper oxygen-fuel mixture for ignition and combustion, efficiency of combustion, and area of exit port |
| Domes | Volume, specific heat and mass of dome and heat sink, heat conductivity |
| Orifices and Injectors | Cross-sectional area and coefficient of discharge injectors |
| Manifold | Orifice data, volume, specific heat and mass of manifold and heat sink, heat conductivity |
| Heat Exchanger | Specific heat and heat conductivity of exchanger, wetted area, and geometry |

The theoretical and experimental studies performed on the Apollo-Saturn 200 and 500 series vehicles should be provided. These data will be used for the following purposes:

- To avoid the possibility of duplicating effort
- To provide a check on the simulation
- To provide a means of restricting the size of the simulation
- To provide expected inputs for a given simulation

The particular areas of interest with respect to the above studies are as follows:

- Vibration Analyses
- Heat Transfer Analyses
- Propellant Flow Analyses
- Ullage Pressure in the Fuel and Oxidizer Tanks
- Acceleration
- Thrust
- Actuation Times
- On-off Sequences

- Time Delays
- Moment of Inertia
- Center of Gravity Location
- Location of the Various Propulsion Elements with Respect to the Center of Gravity

These studies should be related to a specific Apollo-Saturn vehicle and to time.

The inadvertent omission of hardware items, misinterpretation of functional relations, and improper use of handbook data are a very real possibility in a system as large as the Saturn V; therefore, the following data are needed:

- Complete set of drawings for each of the Apollo-Saturn launch vehicles to be simulated
- Complete parts list for each Apollo-Saturn launch vehicle to be simulated
- Functional descriptions and schematics at each required level of detail
- Physical characteristics of the various materials, fluids, gases, parts, and assemblies as a function of anticipated environments
- A description of the various identifiers (specification numbers, drawing numbers, ECP numbers, CEI number, SCN numbers etc.) as they relate to the program, stage, functional subsystems, subassemblies, and component parts

#### 3.1.1.2 Processing

As pointed out in paragraph 3.1.1.1, the data base for LVCLS will be functional rather than hardware oriented. This means that hardware information such as parts lists, wiring lists, etc., must be translated into logical expressions, differential equations, etc., and inserted into the data base in the prescribed format. This then leads to the conclusion that a functional component is not necessarily synonymous with a hardware component. By treeing out the systems, a partial list of functional components consists of each of the following:

- Valves
- Pipes
- Pipe Connectors
- Orifices
- Turbo Pumps
- Heat Exchangers
- Regulators (flow, pressure)

- Tanks, Gas Bottles, Ignition Chambers
- Manifolds
- Injectors
- Timers
- Pressure Switches
- Gates
- Flow Meters
- Electric Motors
- Reducing Flanges
- Venturis
- Domes
- Headers
- Combustion and Ignition Chambers
- Ignitors
- Nozzles

Each of the functional components will be described by the data base representation of differential equations, logical statements, and dynamic criteria which may affect change of state. This description will be complete and will also include connectors to relate the functions of each of the system components to those of the adjacent ones.

In addition, more than one expression will be required to describe a quantity. For example, consider the problem where the pressure at the outlet side of a valve is desired. Further, the valve is closed until it receives a signal from a timer at which time it begins to open. The pressure at the outlet side is equal to the inlet pressure less the pressure drop in the valve. The pressure drop in the valve is a function of the flow rate, impedance to flow, temperature of the effluent, and time. Until the timer issues the signal (status = 0), the impedance is infinite. After the signal has been issued (status = 1), the impedance decreases as a function of time until the valve portion is at its limit. Thus, the expression to be used for calculation of impedance depends on the status of timer output.

The LVCLS must recognize another point—the dynamic criteria for a change of state for a functional component can be dependent on the present status of that component. For example, consider a relief valve which is set to maintain a pressure between 37 and 40 psia. If the valve is closed (status = 0), it will start to open (changing to status = 1) when the pressure rises to 40 psia. If the valve is open (status = 1), it will start to close (changing to status = 0) when the pressure drops to 37 psia.

A condition whereby the status of an equipment is a function of both discrete and dynamic activities must be recognized. For example, consider the ignition phase timer (IPT) in the J-2 engine. It starts counting down upon receipt of a signal that the start tank discharge timer has run down. The output status from the IPT, when it has run out, instigates either the cut-off sequence (status = 0) or continuation of the start-up (status = 1).

Status (1) requires the two following conditions:

- Signal from ASI monitor that ignition has taken place (discrete)
- Fuel manifold temperature is at or below a given critical valve (dynamic).

Processing of discrete-dynamic simulation will require the following functions, exclusive of input-output and storage and retrieval:

- Logic Equation Processor
- Differential Equation Processor
- Function Evaluator.

The network to be operated on by the differential equation processor will be determined by the status of the discrete variables. As the status of these variables changes, the network is modified. These data are determined by the logic equation processor for such activities as operator actions, time delay actions, etc., and by the function evaluator to account for such activities as pressure, flow, or current criteria equalled or exceeded. The results from both the logic equation processor and the function evaluator appear in a table. Thus the fourth item required is:

- Status Table.

The LVCLS must be capable of performing a discrete simulation alone as well as a combined discrete-dynamic one. This will require the following:

- Additional logical expressions to account for operator activities
- Additional logical expressions, with associated time delays, to account for delays normally due to dynamic responses. Use of these additional expressions will replace normal use of differential equations and dynamic criteria
- Additional connectors which apply to discretes only

The results from such an analysis will be a listing of component status versus time. The formats of outputs desired by the user will be accomplished by the output editor along with the output routines. They will give one of the following:

- A list giving component states versus time but will not include variable values versus time (output B, Table 4-1)
- A list of status changes which occur in specified components due to a stimulus such as operator activities (output C, Table 4-1).

By combining the status information with a comparison, an output editor, and an output routine, the LVCLS will furnish the following:

- A list of components which changed state at some time different from that indicated by an input list. Only the differences with their times will be listed (output K, Table 4-1).

An overall diagram of the basic logic required for tying together the discrete and dynamic portions of LVCLS is given in Figure 4-2. The computer flow diagram for the LVCLS is provided in Figure 4-3. The flow of the discrete and dynamic program modules is shown in Figure 4-4.

3.1.1.3 Outputs

The outputs which will be provided by the Launch Vehicle Component Level Simulation are noted in Table 4-1. These outputs are specified in Statement of Work for Contract NAS 8-20060, Modification No. 1.

The Output Monitor will edit the simulation output, organize it into proper format, and provide the engineering outputs in the form of lists, plots, or displays. The general form of each output is indicated in Table 4-1.

The Output Monitor will be programmed to provide a standard set of outputs for each identified use of the component level simulation. The relationship between Applicable Uses and Outputs is depicted in Table 4-2. The desired use or uses for a simulation will be indicated via the Input Monitor and the standard outputs for that use or uses will be provided. Similarly, the equipment identifiers, parameters, and other data of normal interest to a specific simulation will be automatically selected for these outputs. A listing of the representative data elements versus each output is provided by Table 4-3. A manual select mode will be provided such that the user may modify the output to fit specific needs.

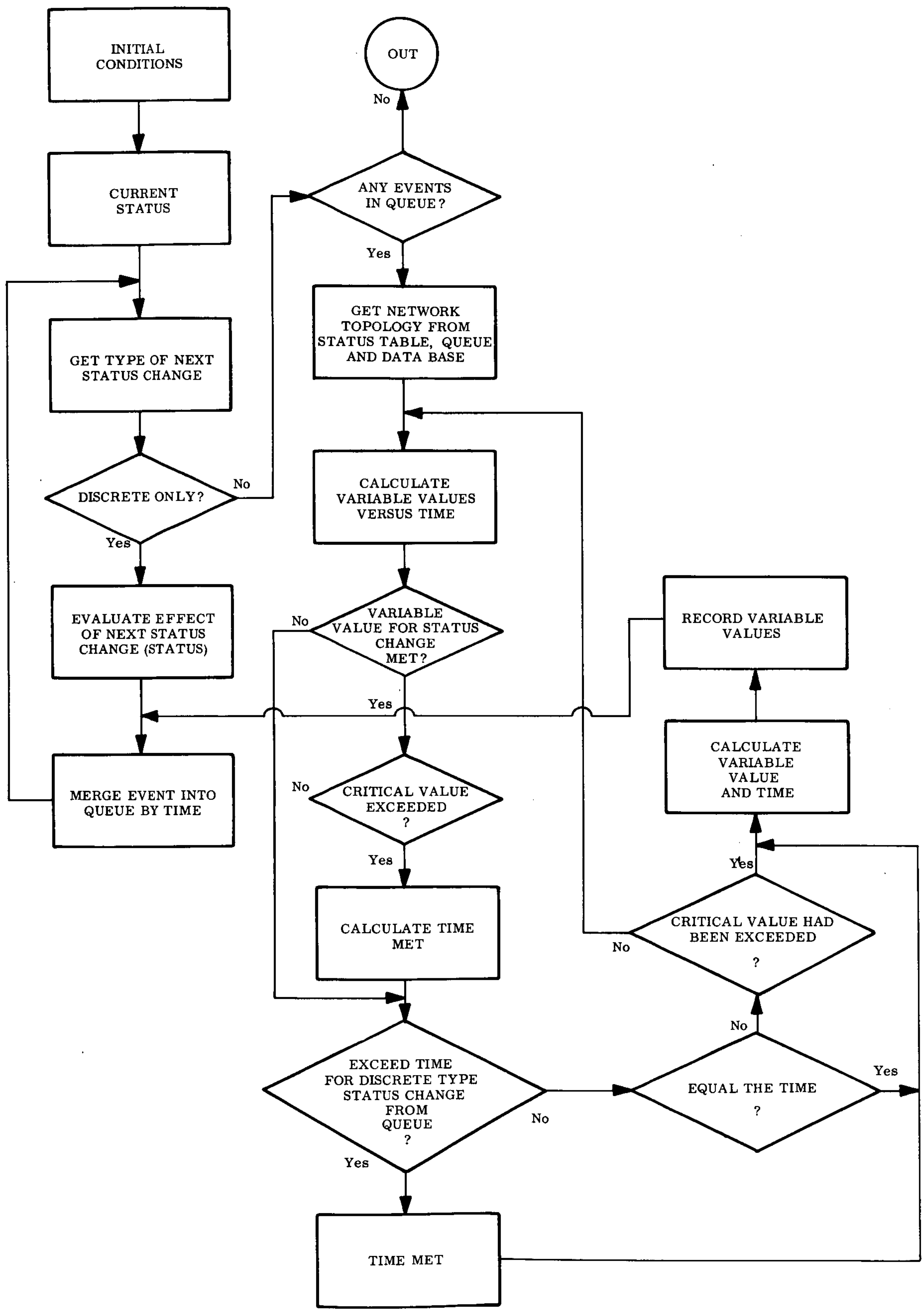



Figure 4-2. Discrete/Dynamic Coordination Schematic

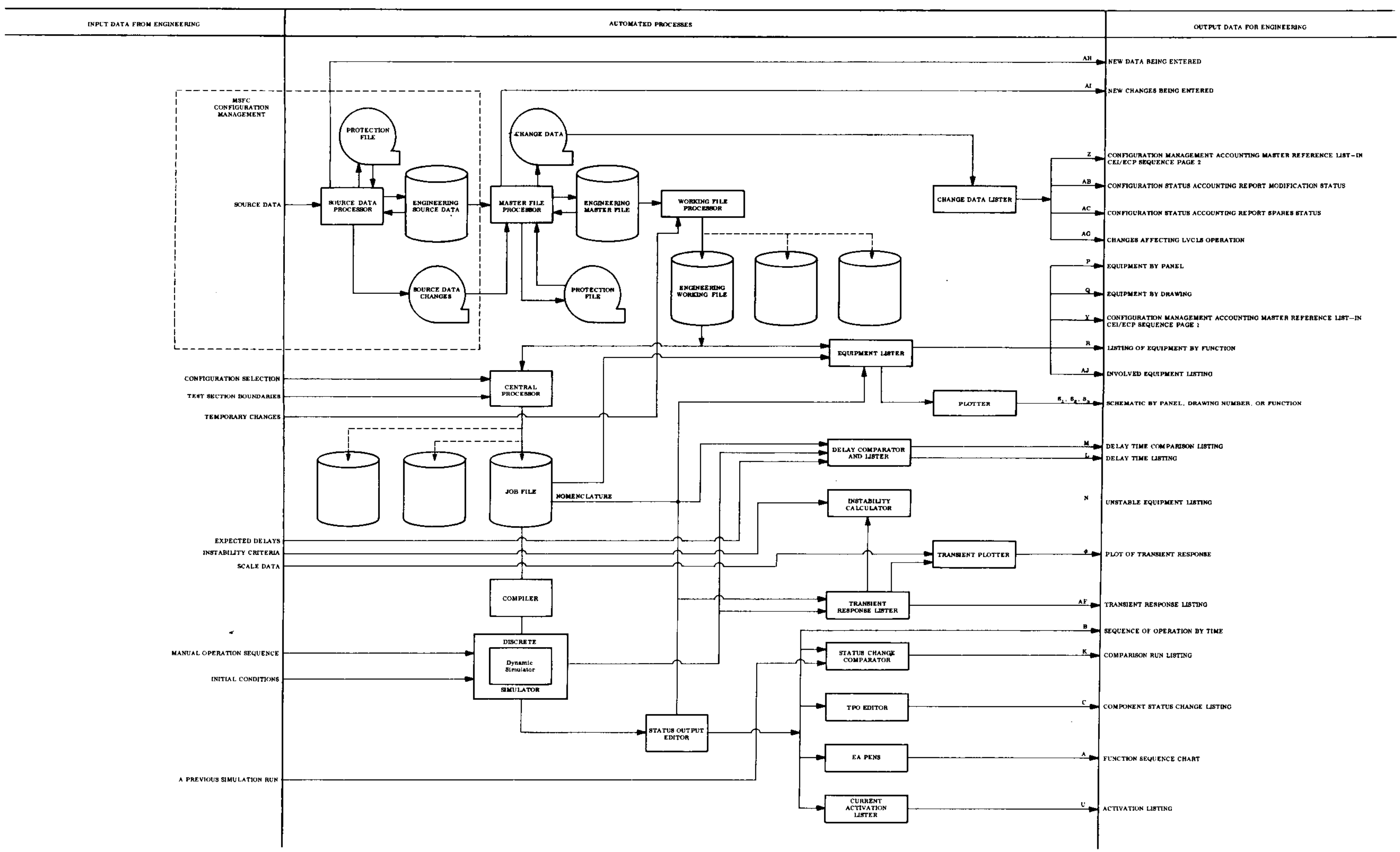



Figure 4-3. Flow Diagram of the LVCLS System

The Simulator Control reads and stores the initial conditions, the sequence, the times of operator actions, and the equations from the Compiler. It processes the compiled equations by selecting either the Discrete Processor or the Dynamic Controller as needed.

The Dynamic Controller selects the appropriate equation processor the results of which are compared with limits and previous values. The Discrete Processor evaluates the Discrete Logic. The results are compared with previous status values.

The Simulator Control continues to select the discrete and dynamic sections until a component changes status. It processes the remaining equations until the earliest status change is determined and its time of change. This time and the entire status table are output for the Status Output Editor program. The program continues to process the operator activations and equations until all status changes have taken place.

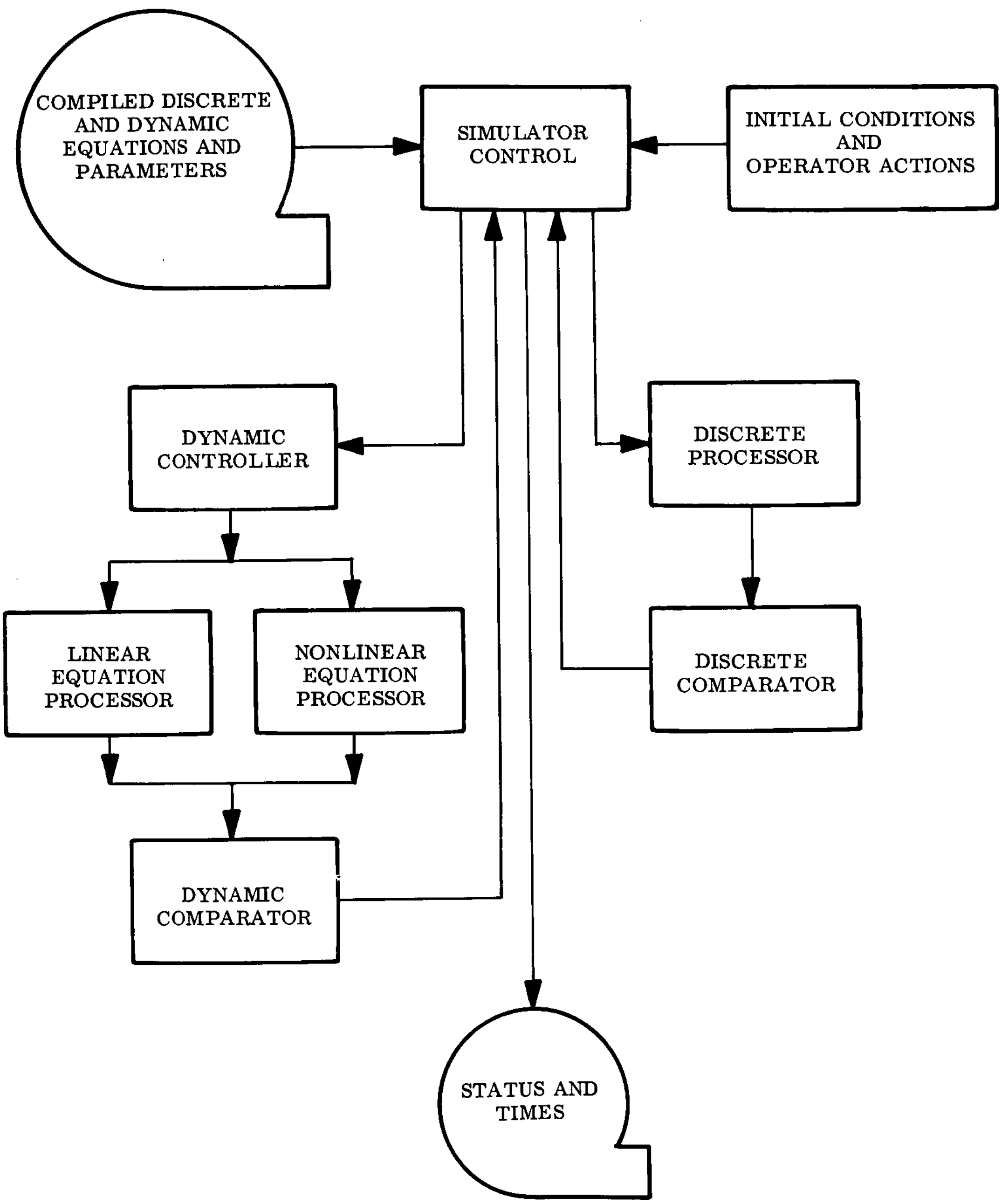



Figure 4-4. Flow Diagram of the Discrete and Dynamic Program Modules

Ta

Uses Ve

| Applicable Uses | A | B | C | K | L | M | N | O | I |
|---|---|---|---|---|---|---|---|---|---|
| II Keep track of approved change orders, drawing changes, and hardware changes made in the simulation data file and the resultant configuration | | | | | | | | | |
| III Insert approved changes into the central data file | | | | | | | | | |
| V Calculate expected times for events of a selected portion of the launch vehicle and ground support systems | | | | | X | X | | | |
| VI Perform transient analysis of a selected portion of the launch vehicle and ground support systems | | | | | | | X | X | |
| VII Follow signals through a selected portion of the vehicle on a discrete basis | X | X | X | X | | | | | |
| VIII Relate the simulation to the racks, equipment numbers, etc., as given on panel schematics, interconnection diagrams, and advanced schematics | | | | | | | | | |
| XIII Allow a user to set up conditions which identify a portion of a proposed actual checkout or countdown sequence | | | | | | | | | |
| XV Define and keep track of equipments which have been activated with time or number of activations | | | | | | | | | |
| XVII Compare resulting sequences with desired ones for checkout or countdown activities listing for comparison run | | | | X | | | | | |
| XIX Configuration management documentation data center and control | | | | | | | | | |



FOLDOUT FRAME

2

ble 4-2

rsus Outputs

| Outputs | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Q | R | $S_1$ | $S_2$ | $S_3$ | U | Y | Z | AB | AC | AF | AG | AH | AI | AJ |
| | | | | | | | | | | | | X | | | |
| | | | | | | | | | | | | | X | X | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | X | | | | |
| | | | | | | | | | | | | | | | |
| | X | X | X | X | X | | | | | | | | | | |
| | | | | | | | | | | | | | | | X |
| | | | | | | X | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | X | X | X | X | | | | | |

FOLDOUT FRAME

T

Data Requirement

| Data Elements | A | B | C | K | L | M | N | O | P |
|---|---|---|---|---|---|---|---|---|---|
| Equipment Location in Vehicle | | | X | X | X | X | X | X | X |
| Equipment Drawing and Page Number | | | X | | | | | | X |
| Equipment Specification Number | | | | | | | | | |
| Equipment CEI Numbers | | | | | | | | | |
| Equipment ECP Numbers | | | | | | | | | |
| Equipment SCN Numbers | | | | | | | | | |
| Equipment Part Numbers | | | X | | | | | | X |
| Equipment Nomenclature | | X | X | | X | X | X | X | X |
| Equipment Quantity | | | | | | | | | |
| Contractor Number | | | | | | | | | |
| Contractor Identification | | | | | | | | | |
| Change Title | | | | | | | | | |
| Equipment Geographic Location | | | | | | | | | |
| Contract End Item Nomenclature | | | | | | | | | |
| Serial Numbers Involved in an ECP | | | | | | | | | |
| Issue Date of ECP | | | | | | | | | |
| Contract Change Notification Number | | | | | | | | | |
| Related Documents to an ECP | | | | | | | | | |
| Configuration Control Board Numbers | | | | | | | | | |
| Identification of Critical Component | | | | | | | | | |
| Identification of Agency or Contractor Who Prepared Specification | | | | | | | | | |
| Identification of Specification Custodian | | | | | | | | | |
| Specification Scheduled Issue Date | | | | | | | | | |
| Specification Actual Issue Date | | | | | | | | | |
| Cognizant R&DO Laboratory | | | | | | | | | |



FOLDOUT FRAME

2

able 4-3

s Versus Outputs (Part 1)

| Outputs | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Q | R | $S_1$ | $S_2$ | $S_3$ | U | Y | Z | AB | AC | AF | AG | AH | AI | AJ |
| X | X | X | X | X | X | X | X | X | | X | X | X | X | X |
| X | X | | X | | X | | | | | | | X | X | X |
| | | | | | | X | X | | | | | X | | |
| | | | | | | X | X | X | X | | | X | | |
| | | | | | | | X | X | | | | X | X | |
| | | | | | | | X | X | X | | X | X | X | |
| X | X | | | | X | X | | | | | | X | | X |
| X | X | X | X | X | X | | | | | X | | X | X | |
| | | | | | | X | X | | X | | | X | X | |
| | | | | | | X | X | | X | | | X | X | |
| | | | | | | | X | | X | | | X | X | |
| | | | | | | | X | | | | | X | X | |
| | | | | | | X | X | X | | | X | X | X | |
| | | | | | | X | X | | | | | X | X | |
| | | | | | | | X | X | | | | X | X | |
| | | | | | | | X | | | | | X | X | |
| | | | | | | | X | | | | X | X | X | |
| | | | | | | | X | | | | | | | |
| | | | | | | X | X | | | | | X | X | |
| | | | | | | | X | | | | | | | |
| | | | | | | X | X | | | | | | | |
| | | | | | | X | X | | | | | | | |
| | | | | | | X | X | | | | | | | |
| | | | | | | X | X | | | | | | | |
| | | | | | | X | X | | | | | | | |



FOLDOUT FRAME

T

Data Requiremen

| Data Elements | A | B | C | K | L | M | N | O | I |
|---|---|---|---|---|---|---|---|---|---|
| Project | X | X | X | X | X | X | X | X | X |
| Part Numbers Affected By an ECP | | | | | | | | | |
| New Part Numbers as a Result of an ECP | | | | | | | | | |
| Quantity Affected By an ECP | | | | | | | | | |
| Number of Kits to be Procured by the Contractor | | | | | | | | | |
| Quantity in Production to be Modified | | | | | | | | | |
| Quantity Requiring Mod. by Contractor | | | | | | | | | |
| Quantity of Required Modification by Contractor Completed | | | | | | | | | |
| Quantity Requiring Mod. by MSFC | | | | | | | | | |
| Quantity of Required Modifications by MSFC Completed | | | | | | | | | |
| Date of Kit First Schedule | | | | | | | | | |
| Date Kit Accepted | | | | | | | | | |
| Date Kit Delivered | | | | | | | | | |
| MSFC Report Number | X | X | X | X | X | X | X | X | X |
| Date of Modification Status | | | | | | | | | |
| Operation or Event | | X | | X | X | X | X | | |
| Date of Run | X | X | X | X | X | X | X | X | X |
| Modification Type Code | | | | | | | | | |
| Level of Modification | | | | | | | | | |
| Kit Identification | | | | | | | | | |
| Required Delivery Date | | | | | | | | | |
| Actual Delivery Date | | | | | | | | | |
| Expended Code | | | | | | | | | |
| Function Name | | X | | X | X | X | X | X | |
| Time | | X | X | X | X | X | X | X | |

2

able 4-3

ts Versus Outputs (Part 2)

| Outputs | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P | Q | R | $S_1$ | $S_2$ | $S_3$ | U | Y | Z | AB | AC | AF | AG | AH | AI | AJ |
| X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | | | | | | | | X | X | X | | X | X | X | |
| | | | | | | | | | X | X | | X | X | X | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | X | | | | | |
| | | | | | | | | | | X | | | | | |
| | | | | | | | | | | X | | | | | |
| | | | | | | | | | | X | | | | | |
| | | | | | | | | | | X | | | | | |
| | | | | | | | | | | X | | | | | |
| | | | | | | | | | | X | | | | | |
| | | | | | | | | | | X | | | | | |
| | | | | | | | | | | X | | | | | |
| X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | | | | | | | | | | X | | | X | X | |
| | | | | | | | | | | | | X | X | X | X |
| X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | | | | | | | | | X | | | | X | | |
| | | | | | | | | | X | | | | | | |
| | | | | | | | | | X | | | | | | |
| | | | | | | | | | X | | | | | | |
| | | | | | | | | | X | | | | | | |
| | | | | | | | | | X | | | | | | |
| | | X | | | X | X | | | | | X | X | X | X | X |
| | | X | | | X | X | | | | | X | | | | X |

FOLDOUT FRAME

Ta

Data Requirement

| Data Elements | A | B | C | K | L | M | N | O | P |
|---|---|---|---|---|---|---|---|---|---|
| Equipment On-Off Status | | X | X | X | | | | | |
| Equipment Panel Identification | | | | | | | | X | X |
| Delay Times | | | | | | X | X | | |
| Current vs Time Per Location | | | | | | | | X | |
| Voltage vs Time Per Location | | | | | | | | X | |
| Power vs Time Per Location | | | | | | | | X | |
| Temperature vs Time Per Location | | | | | | | | X | |
| Angle or Displacement vs Time Per Location | | | | | | | | X | |
| Velocity vs Time Per Location | | | | | | | | X | |
| Acceleration vs Time Per Location | | | | | | | | X | |
| Static Pressure vs Time Per Location | | | | | | | | X | |
| Dynamic Pressure vs Time Per Location | | | | | | | | X | |
| Flow vs Time Per Location | | | | | | | | X | |
| Force vs Time Per Location | | | | | | | | X | |
| Phase and Amplitude vs Frequency Per Location | | | | | | | | X | |
| Energy vs Time Per Location | | | | | | | | X | |
| Mass or Volume vs Time Per Location | | | | | | | | X | |
| Mass Distribution vs Time | | | | | | | | X | |
| Torque vs Time Per Location | | | | | | | | X | |
| Open Loop Gain vs Open Loop Phase | | | | | | | | X | |
| Change in Dimension (Length, Width, or Thickness) | | | | | | | | | |
| Change in Volume | | | | | | | | | |
| Change in Mass | | | | | | | | | |
| Change in Mass Distribution | | | | | | | | | |
| Change Resistance (Electrical, Mechanical, Hydraulic, or Thermal) | | | | | | | | | |



FOLDOUT FRAME

2

ıble 4-3

s Versus Outputs (Part 3)

| Outputs | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Q | R | $S_1$ | $S_2$ | $S_3$ | U | Y | Z | AB | AC | AF | AG | AH | AI | AJ |
| | | | | | | | | | | | | | | |
| X | X | X | X | X | X | | | | | X | | | | X |
| | | | | | | | | | | | X | | | |
| | | | | | | | | | | X | | | | |
| | | | | | | | | | | X | | | | |
| | | | | | | | | | | X | | | | |
| | | | | | | | | | | X | | | | |
| | | | | | | | | | | X | | | | |
| | | | | | | | | | | X | | | | |
| | | | | | | | | | | X | | | | |
| | | | | | | | | | | X | | | | |
| | | | | | | | | | | X | | | | |
| | | | | | | | | | | X | | | | |
| | | | | | | | | | | X | | | | |
| | | | | | | | | | | X | | | | |
| | | | | | | | | | | X | | | | |
| | | | | | | | | | | X | | | | |
| | | | | | | | | | | X | | | | |
| | | | | | | | | | | X | | | | |
| | | | | | | | | | | X | | | | |
| | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | X | X | X | |

Ta

Data Requiremen

| Data Elements | A | B | C | K | L | M | N | O | I |
|---|---|---|---|---|---|---|---|---|---|
| Change in Capacitance | | | | | | | | | |
| Change in Inductance | | | | | | | | | |
| Change in Current Capacity | | | | | | | | | |
| Change in Voltage Rating | | | | | | | | | |
| Change in Rated Power | | | | | | | | | |
| Change in Distribution Buses | | | | | | | | | |
| Change in Location | | | | | | | | | |
| Change in Rated Horsepower | | | | | | | | | |
| Change in Damping Coefficients | | | | | | | | | |
| Change in Cross Sectional Area | | | | | | | | | |
| Change in Static Pressure | | | | | | | | | |
| Change in Ullage Pressure | | | | | | | | | |
| Change in Rated Velocity | | | | | | | | | |
| Change in Density | | | | | | | | | |
| Change in Viscosity | | | | | | | | | |
| Change in Emmissivity | | | | | | | | | |
| Change in Rated Torque | | | | | | | | | |
| Change in Clearance | | | | | | | | | |
| Change in Tolerance | | | | | | | | | |
| Change in Specific Impulse | | | | | | | | | |
| Change in Logic Statements | | | | | | | | | |
| Deletion of Equipments | | | | | | | | | |
| Change in Spring Constants | | | | | | | | | |
| Change in Absorptivity | | | | | | | | | |
| Change in Latent Heat of Vaporization | | | | | | | | | |

FOLDOUT FRAME

ble 4-3 2

s Versus Outputs (Part 4)

| | Outputs | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Q | R | $S_1$ | $S_2$ | $S_3$ | U | Y | Z | AB | AC | AF | AG | AH | AI | AJ |
| | | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | | X | X | X | |
| | | | | | | | | | | | | X | | | |
| | | | | | | | | | | | | X | | | |

/

Ta

Data Requirement

| Data Elements | A | B | C | K | L | M | N | O | P |
|---|---|---|---|---|---|---|---|---|---|
| Change in Specific Heat | | | | | | | | | |
| Change in Internal Energy | | | | | | | | | |
| Change in Geometry | | | | | | | | | |
| Change in Coefficient of Expansion | | | | | | | | | |
| Change in Inertia | | | | | | | | | |
| Change in Surface Finish | | | | | | | | | |
| Vehicle Identification | X | X | X | X | X | X | X | X | X |
| Effective Configuration Date | X | X | X | X | X | X | X | X | X |



FOLDOUT FRAME

2

ble 4-3

; Versus Outputs (Part 5)

| | Outputs | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Q | R | $S_1$ | $S_2$ | $S_3$ | U | Y | Z | AB | AC | AF | AG | AH | AI | AJ |
| | | | | | | | | | | | | X | | | |
| | | | | | | | | | | | | X | | | |
| | | | | | | | | | | | | X | | | |
| | | | | | | | | | | | | X | | | |
| | | | | | | | | | | | | X | | | |
| | | | | | | | | | | | | X | | | |
| | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |

FOLDOUT FRAME

Outputs Y, Z, AB, and AC are four of the reports which will be issued by the MSFC Configuration Management Accounting system. The format of these outputs will conform with the formats noted in the report R-COMP-A-66-1, The MSFC Configuration Management Accounting System, dated 5 January 1966. The format of these outputs as indicated in the report, Analytical Study of Launch Vehicle Component Level Simulation, was based on exhibits in NPC 500-1 which are different from those noted in the referenced MSFC report. Also, output Y will present all of the information contained in outputs Y and AA using the NPC 500-1 exhibits. It was noted in paragraph 3.1.1.1 that much of the input data required for these outputs serves no purpose for the other uses and outputs of the simulation.

Schematics by panel, drawing, and function (outputs $S_1$, $S_2$ and $S_3$, respectively) are indicated as outputs in Table 4-1. It is recommended that these outputs be omitted for the reasons outlined in paragraph 3.1.1.1.

Output AG, listing of changes which affect the simulation, is based only on the changed inputs. The effects of the changes on the discrete and dynamic simulation are not evaluated prior to this listing.

### 3.1.2 OPERABILITY

Computational modules of the Launch Vehicle Component Level Simulation (LVCLS) computer program will be written in FORTRAN IV. File manipulation modules will be written in COBOL. The Executive Monitor and submonitors will be written in a more machine-oriented language to increase overall efficiency of running. Since the LVCLS will be operating on a NASA computer at Huntsville, this computer will be considered as the object computer, and coding will be performed so as to use it most efficiently. Final debug and checkout will be performed on the NASA computer at Huntsville. Initial running and debugging may be performed on another computer system, but where there is a conflict which affects efficiency of operation it will be resolved in favor of the computer system with which the LVCLS computer program will become operational at Huntsville.

The LVCLS computer program will be written in modular form so that a particular program module may be withdrawn, rewritten, modified, or maintained without affecting other capabilities of the LVCLS computer program which do not utilize the particular program module. When a particular program module is withdrawn from the LVCLS computer program temporarily, a dummy module will be inserted to return control to the monitor system should the module be called inadvertently. If it becomes desirable to do so, there will be a capability provided to communicate with digital-to-analog and analog-to-digital conversion hardware so that analog computation, analog representation, or vehicle hardware systems may be accomodated at a later date. Although many LVCLS problems will be extensive in size and should be computed by themselves, many others will be small enough to justify time-shared computation. For this reason, capability to operate in a time-shared mode will be included.

The LVCLS computer program will be transportable in card form or magnetic tape form. It will operate in a normal computer facility environment and utilize standard methods of coding including sufficient explanatory comments with each program module, integration routine, or similar subset of instructions so that any experienced computer programmer may follow the logical flow of the program and understand its operation. The machine-oriented language used will be isolated from other instructions and confined to those cases where there is a significant advantage in its use.

#### 3.1.2.1 Error Analysis and Recovery

The automated processes for LVCLS are composed of modular computer programs which provide accessibility for alterations and error corrections.

Each modular program shall be verified by using a known set of input data and comparing the output with the corresponding set of known output data. At the completion of the modular examination, coupled modules will also be examined in identical fashion to demonstrate coupled processing ability.

In order to facilitate the computer program debugging operation for the removal of errors and inaccuracies, special subroutines may be written which have the following characteristics:

a. Input Check

   Input cards shall be identified in a sequential manner and counted. The input cards shall be scanned for inconsistencies and flagged in the module output to facilitate error correction.

b. Universal Dump

   A dynamic universal memory printout will provide an examination of the contents of any addressable registers, working memory, and testable alarms or conditions at the time of the dump. It is designed to assist the programmer in finding bugs as they occur during the testing of the programs.

c. Trace

A trace program will trace the operations executed in a program while executing it and will list all register contents. This is an additional assist in debugging a program.

d. Checksum Corrector

This routine reads a deck of relocatable binary type-3 instruction cards and tests each card for an incorrect checksum. Upon finding a card with an incorrect checksum, this card image is punched out with a corrected checksum field.

### 3.1.2.2 Maintainability

The modular structure and the explanatory comments included in the program modules as mentioned in paragraph 3.1.2 above are design features which support maintainability. It is also necessary to exercise good judgment and proper control after the LVCLS computer program is operational. To insure that the LVCLS computer program is properly supported, an LVCLS configuration management procedure will be written and provided with the program. This procedure will include sequential steps required to evaluate proposed changes to the computer program. It will also include a description of the mechanical methods of protecting the data base and program library together with a designation of persons responsible for and those authorized to physically implement changes and verify program operability and suitability.

When a particular program module is undergoing maintenance, all other program modules shall remain operable to the extent that any problem not directly using the module under maintenance will be computed at its normal speed and accuracy.

### 3.1.2.3 Expandability

The same modular structure which enhances maintainability also provides for expandability and for changeability. To expand the LVCLS computer program, it will be necessary to write the new program module, compile it, and debug it prior to implementing it through the LVCLS configuration management system. The mechanical steps include adding appropriate references to the program module in the monitor system along with its input requirements for computation. This should decrease the possibility of rewriting other program modules.

3.1.2.4 Adaptability

The LVCLS computer program will for the most part be written in FORTRAN IV. Those portions such as the multiple level monitor system which are written in machine-oriented language will be limited in size and programmed so as to facilitate change to another computer system if required. It is anticipated that the Engineering Source Data will be added to the MSFC Configuration Management Accounting system in the same general way as is the current administrative data. The LVCLS computer program will include an Input Processor to take this Engineering Source Data from the random storage on Configuration Management discs and organize it into sequentially linked sets to form an Engineering Master Data File which is efficient from the point of view of providing Engineering Working Files for the desired configurations.

The user-oriented language developed for the LVCLS computer program will utilize engineering terms and mathematical expressions in current use at MSFC. This will necessitate some internal functions responding to more than one user representation and require that the user select the desired output form.

The LVCLS computer program is not expected to be subjected to hostile environments. It is intended to be transportable either on the normal cards or magnetic tape for use in a multiple-use computer facility. The supporting data base will be stored on and used from disc files but may be relocated on magnetic tape when inactive.

3.1.2.5 Transportability

The LVCLS computer program is a stored program intended for use on the MSFC computer system. It is not expected to be transportable in the usual sense. If it becomes desirable to move the program to other facilities, it may be moved easily in the form of magnetic tape or punched cards. It will then be necessary to make some changes in the machine-oriented language portions of the program (e.g., monitor structure) in order to operate on the new computer system. The LVCLS computer program and data base will be extensive in size and require very large computer facilities for efficient operation. The LVCLS computer program will be designed and coded to operate on a computer system having the following capabilities:

- Core memory of 131K words or greater
- Access time of one microsecond with compatible add, multiply, and divide times
- Multiple disc memories each having a capacity of 33 million words
- Time-sharing capability

- Remote terminals two of which have local processors
- Digital plotters
- CRT displays
- High-speed tape, card reader, printer, etc.
- Digital-to-analog and analog-to-digital conversion hardware.

3.1.2.6 Human Performance

The LVCLS computer program is intended for use directly by engineers of the various MSFC laboratories. It will include a user-oriented language which recognizes common engineering terms from each branch of engineering. Specifically, there will be included terms for electrical and electronic problems, mechanical problems, compressible flow of fluids, thermodynamics, aerodynamics, ballistics, etc. Many of these user requirements will translate into identical internal computations and manipulations. The maintenance, updating, configuration control, etc. which relate directly to the program and its engineering data base will not be performed by the using engineers but rather by skilled computer system programmers assigned to the task by the MSFC computer laboratory. Where changes are required by the user for engineering evaluation, they will be of a temporary nature. These temporary changes may be preserved in the form of output if required.

## 3.2 CEI DEFINITION

### 3.2.1 INTERFACE REQUIREMENTS

The LVCLS computer program will be utilized directly by engineers. The user-oriented language discussed under paragraph 3.1.2.6 is therefore an interface requirement. Likewise, the computer system requirements discussed under paragraph 3.1.2.5 are a restriction. All program modules within the LVCLS will be self sufficient but designed to operate as a portion of the complete simulation. This necessitates that existing program modules which may later be added to the LVCLS computer program library be checked for conflicts such as the use of indices, terms, dimensions of quantities, etc. All modifications of the LVCLS computer program must be controlled under its own LVCLS Configuration Management procedure discussed in paragraph 3.1.2.2 to insure evaluation of proposed changes, file protection, program protection, and subsequent efficient and accurate computation.

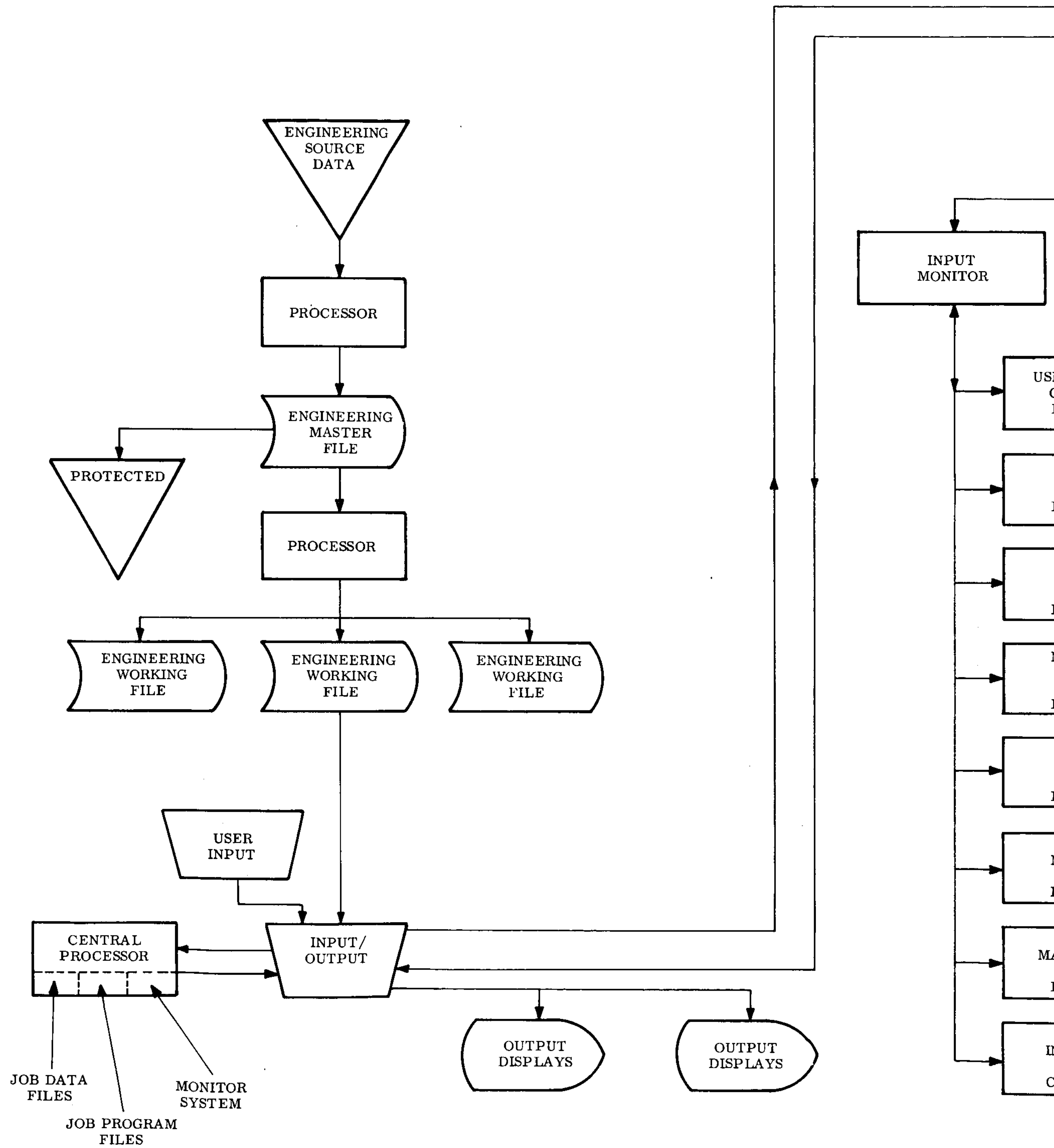
#### 3.2.1.1 Schematic Arrangement

A general discussion of the simulation system was given in paragraph 3.1. A schematic representation of the major elements of this system is given by Figure 4-5.

The Engineering Source Data File will contain a collection of engineering, schedule, cost, and other data for the configurations of the various Apollo-Saturn missions. These data will be stored on this file in a random fashion without regard to configuration or the type of data. The data applicable to the LVCLS will be extracted from this file by a processor. This processor will organize the data and assemble a single listing Engineering Master File. This file will serve as a single point of reference for each unique data element used in the simulation. A series of Engineering Working Files, one for each vehicle configuration, is compiled from the Engineering Master File via a processor. Therefore, the Engineering Working Files will list a given data item in all applicable configurations (multiple listing). All of these data files are limited-access protected files.

The user input data will provide the effective configuration date, vehicle designation, equipment boundaries, temporary change data, and other information pertinent to a specific simulation.

Paragraph 3.1 describes the relationships between the Executive Monitor, Input Monitor, Job Monitor, and Output Monitor for a typical computer run.


ENGINEERING SOURCE DATA
PROCESSOR
ENGINEERING MASTER FILE
PROTECTED
PROCESSOR
ENGINEERING WORKING FILE
ENGINEERING WORKING FILE
ENGINEERING WORKING FILE
USER INPUT
CENTRAL PROCESSOR
INPUT/ OUTPUT
OUTPUT DISPLAYS
OUTPUT DISPLAYS
JOB DATA FILES
JOB PROGRAM FILES
MONITOR SYSTEM
INPUT MONITOR

2

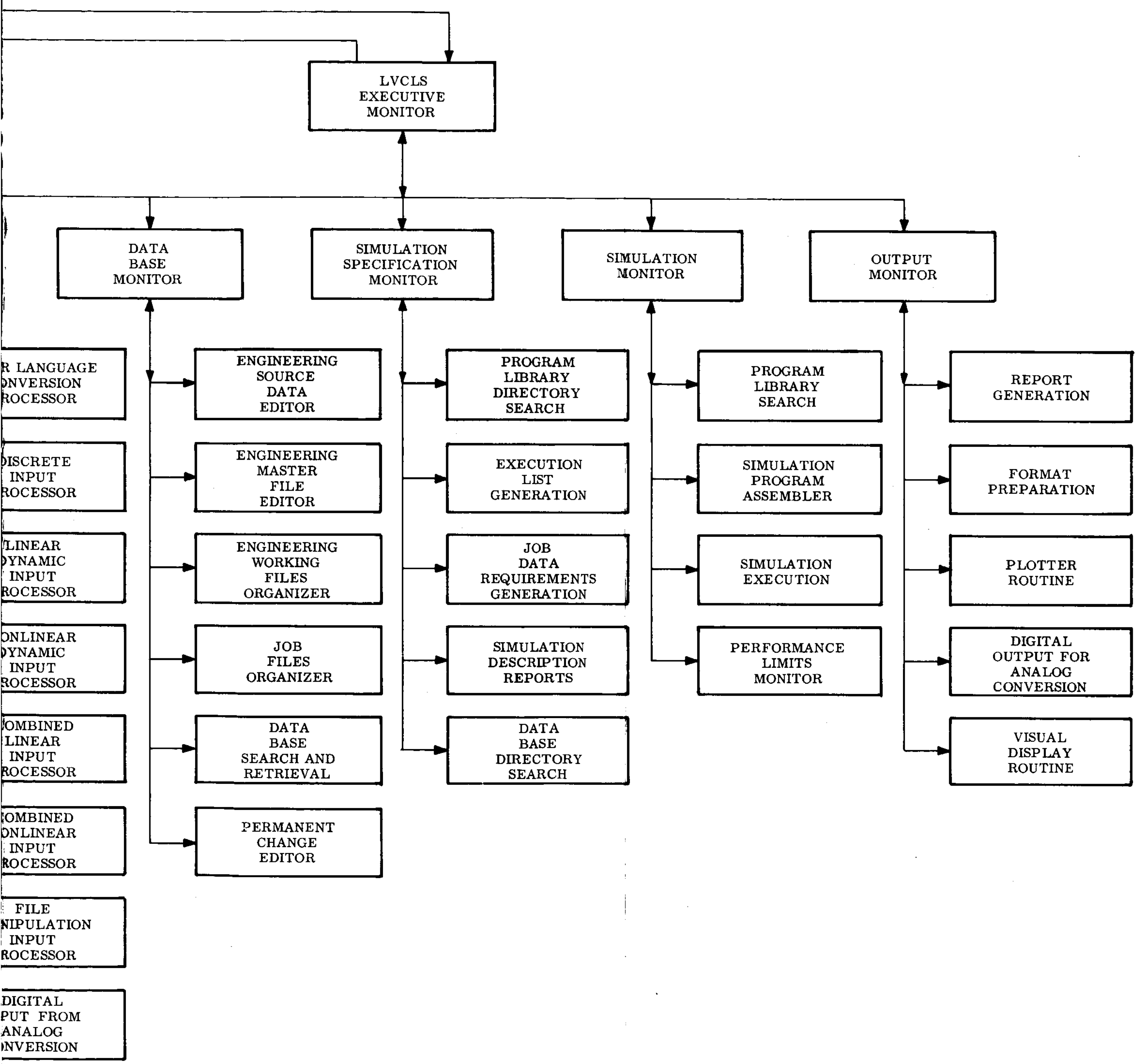



Figure 4-5. Schematic Arrangement of Major Elements of the LVCLS System

#### 3.2.1.2 Detailed Interface Definition

The LVCLS computer program will be designed to operate efficiently on the third-generation computer system currently being selected for the MSFC computational facilities (R-COMP). It will operate under the computer installation monitor system and below this under its own LVCLS monitor system. Computational program modules such as the Linear Dynamic program, the Discrete program, etc., will be written in FORTRAN IV. Data handling program modules such as the Engineering Master File Processing program, the Job File Processing program, etc., will be written in COBOL. The LVCLS Executive Monitor will be written in machine-oriented language. Initial development will be done on an IBM 7044 system with 1301 disc files. Some reprogramming may be required when R-COMP has taken delivery of their new computer system. There will be available Engineering Working Files for three configurations at any one time on discs operating through the input-output for input to the job file which will be partially in core and partially on disc. The working file for each configuration is estimated at 33 million words of 36 bits each. Working files for configurations other than the three in current use will be stored off-line on tape or disc. The job file will vary from 15 thousand words to perhaps as high as four million words depending on the problem. In general, the smaller data base will be accompanied by more complex calculations while the larger data base will be accompanied by simple discrete or binary calculations.

The LVCLS computer program will be capable of operation directly by engineers who will communicate with the simulation through a user-oriented language developed for the purpose.

Communication between the user and the simulation will be in engineering terms normally used in each engineering field such as mechanical, electrical, aerodynamic, etc. Functional representation may be in terms of another single form on the computer.

Data transfer rates should be over 100KC for sequential listings. Memory speed should be 2 microseconds or better.

### 3.2.2 GOVERNMENT FURNISHED COMPUTER PROGRAMS

Other than the computer system monitor discussed in paragraph 3.2.1.2 above there are no Government Furnished Computer programs. The Engineering Source Data will be provided in a manner similar to that specified in report R-COMP-A-66-1 for

The MSFC Configuration Management Accounting System. The computer program which provides this data to LVCLS in disc form is not a component part of LVCLS but is necessary for its operation.

## 3.3 DESIGN AND STRUCTURE

### 3.3.1 GENERAL DESIGN FEATURES

The LVCLS computer program will consist of a number of program modules organized under an Executive Monitor. The modules may be used in many different combinations to perform simulations. The overall LVCLS capabilities are given in the December 1965 report Analytical Study of Launch Vehicle Component Level Simulation. The capabilities included within this particular requirements specification are given in Appendix A of Modification No. 1 to Contract NAS 8-20060. The LVCLS Executive Monitor and submonitors will remain in core during problem solution together with those program modules which are required for the particular problem. Small detailed problems such as the ST-124M X-loop will occupy from 10.5K to 25K of core including data base and programs. Most problems in terms of frequency of use will occupy less than 131K of core; however, a discrete simulation of a complete stage, such as the S-IVB, and its GSE is expected to require disc storage for rapid access to some 1.3 to 4 million words of data in the Job File.

### 3.3.2 DATA BASE REQUIREMENTS

Each Apollo Saturn vehicle configuration is expected to fill disc storage in the amount of some 33 million 36-bit words. These words are organized in hardware and computer program related groups and sequences. Certain words will be listed multiply to reduce data access time during computational computer runs. Several other similar files, one for each active configuration, make up the Engineering Working File. These data are all derived from an Engineering Master File which combines data from various configurations but does not utilize multiple listing for individual common items. The Engineering Master File is itself on disc storage files. The engineering data base for LVCLS will include the functional description of all vehicle and ground support systems for each Apollo Saturn vehicle configuration. This data is composed of logical expressions and actuation criteria, all connectors, dynamic equations at the lowest desired hardware level for each system, and hardware identification.

A list of classes and examples of data input are given in paragraph 3.1.1.1. The size of the data base necessitates utilization of multiple DS-25 class disc storage so that more than one configuration may be available for use on the computer at one time. Each full capacity DS-25 disc file can provide storage for one vehicle configuration. Data base handling and storage will be done by protected magnetic tape and disc files in the computer complex at MSFC-R-QUAL.

Within the data base, the mechanics of entering numerical values of the parameters into the dynamic equations and the process of changing those values can best be described by considering, as an example, a portion of the system in which the analytical determination of the transfer function method is used. The same procedure, with small modifications, will be used for the other methods.

In the system simulated, certain groups of components which may be called subsystems will always function together, regardless of the configuration of the system as determined by the particular use under consideration and the state of the discrete portion of the system. The equations for each subsystem, in terms of component parameters, will be stored. There will also be stored a routine for determining the subsystem transfer function from these equations. When the numerical values of the component parameters are known and have been entered, the transfer function routine will be called upon to determine the subsystem transfer function, which will be stored. From this point on, whenever the subsystem is to be used in a simulation, its transfer function determined by this procedure will be used rather than its basic equations.

Whenever a change in parametric values is to be made, the transfer function routine will be called upon again to take the subsystem equations with the new values and to determine a new subsystem transfer function which will be stored and used in place of the old one.

With this procedure, it will not be necessary to revert to the basic equations for every simulation, and greater efficiency will be achieved. Yet, changes can be readily accommodated.

### 3.3.3 COMPILER/ASSEMBLER UTILIZATION

Reference to the use of compiler/assembler have been made in paragraph 3.1.2 OPERABILITY and in paragraph 3.2.1.2 Detailed Interface Definition. This last designated reference pointed out that the compiler/assembler usage would be as follows:

- Computational Modules in FORTRAN IV
- File Manipulation Modules in COBOL
- Monitors in Assembly Language

This plan has the following advantages:

- Reprogramming problems incurred by applying LVCLS to other computers may be eased
- It may be easier to verify the algorithm during the pilot implementation.

However, when the ultimate machine configuration is chosen, this whole question must be re-opened. The first factor contributing to this statement covers programming efficiency, and this could even lead to the final program being written in assembly language. Another factor concerns the languages which will become available when the final computer system has been selected.

### 3.3.4 EXECUTIVE MONITOR AND CONTROL

The Executive Monitor is not designed to process data or to produce output. Its function is to select and call programs which themselves process the data and produce the output. For many outputs the Monitor will need to call only one program. However, in most instances, the input for this program will have been produced by a series of previously called programs. In all probability these programs would have been called and processed individually and at different times to take advantage of the system's modular construction.

The type of function performed by the Monitor is illustrated by the following example of a data flow path through the system. There are also many other combinations of flow controlled by the Monitor.

The Monitor is told through control cards to produce a "Sequence of Operations by Time" output starting with the random engineering data presented to the card reader. The Monitor, through chain and link structuring, causes the following actions.

The Source Data Editor program is called from the system library and is given control. It reads the data cards and enters the information on the random Source Data File. Control is returned to the Monitor which then calls the Master File Editor. This Editor reads the Source Data File, structures the information, and enters it on the Master File. When control is returned, the Monitor calls the Working File Organizer. This program reads an input control card, selects the desired configuration data from the Master File, and produces the Working File. The Job File Organizer is called next. It reads input data cards for the test section boundaries and temporary changes in the data. It selects the desired data from the Working File, makes the temporary changes, and produces the Job File. When control is returned to the Monitor, it calls the Compiler.

The Compiler, through use of the different input processors, produces a tape containing the compiled equations and returns control to the Monitor. The Monitor then calls the Simulator program which reads data cards for the initial operating conditions and the sequence of operator actions, processes the data and the compiled equations from the tape, and produces the requested "Sequence of Operations by Time" output.

The Executive Monitor, as previously shown, assigns control to the individual programs and assumes control after they have been executed. The sequence and number of program executions is defined by the Monitor from the operator's choice of a beginning point and the desired output.

A previous discussion of the Monitor appeared in paragraph 3.1, PERFORMANCE.

## 4.0 QUALITY ASSURANCE PROVISIONS

### 4.1 PHASE 1 (VERIFICATION) TEST REQUIREMENT

The LVCLS computer program shall be exercised with its data base to produce at least one of each required output type for each applicable use. In addition, the Executive Monitor structure will be exercised to set up and compute a nonlinear dynamic launch vehicle system problem including performance limits such as a static test of the S-II stage with all J-2 engines igniting and guidance input to the actuators. The Executive Monitor structure will also be exercised to set up and compute a linear dynamic launch vehicle system problem such as the small angle equations of the guidance and control system under test stimuli. The discrete program module will be exercised during an automatic countdown such as that of the S-IC stage of the Saturn 501 vehicle from T = -1.86 to liftoff. The LVCLS computer program is not intended to operate with all program modules functioning at one time. It will, therefore, be necessary to test each major combination of program modules which will in fact be expected to function concurrently.

### 4.2 PHASE 2 (INTEGRATION) TEST REQUIREMENT

Since LVCLS computer program is not a computer-based system contract end-item, all test requirements will be included in Phase 1 above.

## SECTION 5

## DYNAMIC SIMULATION APPROACHES

### 5.1 METHODS USED AND PROBLEMS SOLVED

Several methods of solving the dynamic equations of the Launch Vehicle Component Level Simulation were described in Reference 1. It was recognized that, to attain an overall optimum, various portions of the system might best be treated by different methods. In that report, it was recommended that no single method be adapted to the exclusion of the others, but that different combinations be used for different portions of the system.

Work on dynamic simulation techniques during the present contract extension confirms the conclusion reached and substantiates the recommendation. With the different advantages for each method and with the various trade-offs that are possible, the optimum system simulation can be obtained only with flexibility in choice of method for each portion.

The methods studied were:

a. Direct solution by numerical integration of differential equations
b. Analytical determination of system transfer function from subsystem or component transfer matrix
c. Numerical determination of system transfer function by root finding and curve fitting
d. Calculation of system time response by number series from subsystem or component time responses.

In order to compare these methods, they have been tried on a representative portion of the Inertial Measurement Unit. The portion selected was the X-loop of the ST-124-M Inertial Platform as set up for test with a voltage step or impulse input. The equations and numerical values used were obtained from Reference 2. Since three of the four methods studied are primarily applicable to piecewise linear systems, the small angle equations were used. To avoid unnecessary complications, friction, stiction, and deadband were neglected. The torque limiter was used in some solutions.

Details of the application of each method are given in the appendices.

## 5.2 DESCRIPTION OF SYSTEM SIMULATED

The system on which the trial simulations were made was the X-loop of the ST-124-M Inertial Platform, described in the report above. The system configuration is as shown in Figures B-1 and B-2.

The "exact" solution of the equation of the system had already been obtained in the following manner for use in checking the analog runs described in the report. First, the transfer functions $E_o/E_p$ and $E_o/E_A$ were found algebraically from Figure B-2 in terms of the network impedances. Then, from these, the torquer, and the gyro transfer functions, the overall system transfer function, $\beta/E_p$, was found from Figure B-1, also algebraically. The rest of the solution was obtained in steps with the aid of the General Electric time-sharing computer in New York.

Substitution of numerical values and evaluation of the transfer functions in unfactored form was achieved by use of a program for adding and multiplying polynomials. With another program, roots of the various polynomials were found so that they could be expressed as products of linear and quadratic factors. The three transfer functions were expanded in partial fractions with a third program. Finally, the time response for $\beta$ was calculated with a fourth program which evaluated and combined the exponential and trigonometric equivalents of the partial fraction expansion of $\beta/E_p$. All calculations were made to nine significant figures. Since in this process there were four printouts, it cannot be assumed that the end result is accurate to the nine figures given.

## 5.3 COMPARISON OF METHODS

The comparison of the various methods, given here, is based on the work that has been accomplished to date. It must be recognized that there are a number of trade-offs that can be made between accuracy and computer running time that will change the order in which the four methods are listed. These will be discussed below.

The following comparison is made for the ST-124-M gyro problem, treated as a linear system.

a. Accuracy

- Analytical determination of transfer function (>99.99998%)
- Numerical integration (>99.99996%)
- Numerical determination of transfer function (>99.95%)
- Number series (>99%)

b. Computer running time
   - Numerical determination of transfer function (2/3 minute)
   - Analytical determination of transfer function (1 minute)
   - Numerical integration (2-3/4 minutes)
   - Number series (3 minutes)

c. Computational stability (This comparison is made only with regard to conditions that might arise with injudicious use of a particular method.)
   - Analytical determination of transfer function
   - Number series
   - Numerical integration
   - Numerical determination of transfer function

d. Storage Requirements (36-bit words)
   - Analytical determination of transfer function (11,227)
   - Numerical integration (12,388)
   - Numerical determination of transfer function (13,837)
   - Number series (25,152).

## 5.4 DISCUSSION OF COMPARISON AND TRADE-OFFS

As stated, there are a number of factors and trade-offs that can be effected that may change the order of listings above. Consider these for each method in turn.

### 5.4.1 ANALYTICAL DETERMINATION OF TRANSFER FUNCTION

For the gyro problem and under the conditions run, this was the most accurate of the methods, although not by much. Comparing coefficients of the transfer function polynomials with those determined on the time-sharing computer by the method described in Section 4, the maximum difference between any pair was two units in $10^7$. This is well within the possible difference in round-off errors in the two methods due to the different number, kind, and sequence of arithmetic operations performed on the input parameters. It is debatable which should be considered the "exact" theoretical result.

As to computational speed, the analytical method was exceeded only by the numerical transfer function method. But the difference was insignificant. A small gain in the analytical method could be obtained by computing to fewer significant figures, but for a problem like the gyro system, this is not worthwhile. On the other hand, for other problems in which there are many more nonzero coefficients in the equation, it is possible for the computer running time to increase tremendously. It is unlikely that the computational speed for the numerical transfer function method would increase as much, since standard matrix inversion techniques are available for it as discussed in the appendices. Thus, the advantage could swing to the numerical method.

On stability, there is no question that this method is computationally stable; however, for other problems, it is possible that differences between nearly equal large numbers will be taken so that small errors in the input data can result in large changes in the output. This is possible, but highly improbable. If it does occur, the computation can be made in double precision at a sacrifice in computational speed. For the gyro problem, this amounted to an increase of two and a half times.

While this method shows up best with regard to storage requirements, the difference between it and the next two is not significant. The relative order might well be changed if the first three methods were evaluated on a larger or smaller problem.

### 5.4.2 NUMERICAL INTEGRATION

This stood second in accuracy, by a very small amount, but next to last in speed, also by a small amount. For this problem, there is not much difference between this method and the analytical determination of transfer function method with regard to accuracy or storage requirements. Neither method can gain significantly in computational speed by sacrificing accuracy, but both can give increased accuracy at a sacrifice in speed, the one by using a tighter error criterion in the integration step control and the other by using double precision. For the gyro problem, the running time appears to increase faster for the numerical integration method than for the other for a comparable gain in accuracy. This may not be true for other problems, nor may the comparisons be the same.

It is possible that this numerical integration procedure can become computationally unstable with some combinations of input parameters and criteria since most numerical integration schemes do, but not enough runs have yet been made to find the threshold.

Although second in storage requirements, the difference between this and the analytical method is not significant.

### 5.4.3 NUMERICAL DETERMINATION OF TRANSFER FUNCTION

This method comes third on the list of accuracy comparisons on the basis of the runs which have been made. These have, however, shown what can be done to increase the accuracy of the results. In particular, inversion of the regression matrix by partitioning is expected to make a significant improvement.

It is also probable that the partitioning technique will result in a longer computing time so that this method will no longer occupy first place, which it does by the barest of margins. How much of an increase might take place is hard to predict, but whatever it is, the scales might again be reversed for more complicated problems.

In some of the earlier trials made with this method, it was found that incautious selection of some of the parameters and criteria of the procedure could lead to finding incorrect and extraneous roots. It was for this reason the method was placed last in the stability list. It was also proven that this instability could be corrected, and further work may lead to a revision of the ordering in this list.

This method stands third in storage requirements, but it is not greatly behind the other two. However, adding the partitioning feature to the matrix inversion procedure will undoubtedly increase this difference.

### 5.4.4 NUMBER SERIES

Due to difficulties in getting the computer program to operate, this method does not show up as well as the others. Further work will result in improvements, just as in the cases of other methods. This method stands last in accuracy, but no concerted effort has yet been made to study ways of reducing the propogation of errors. A simple network problem was tried, in which the time interval was picked to be one-fifth the shortest constant, as recommended in the published literature, and from this, an estimate was made of the probable accuracy of the number series method of solving the gyro problem. It was obvious that the time interval should be shorter to get better accuracy.

The computer running time for the gyro problem was estimated from the time for a simplified problem, and may be too high. On the other hand, reducing the time interval to improve the accuracy will result in increased running time because of calculating more points in the series. It is not expected that much improvement can be made in running time for this method, except for systems which are nearly linear in time.

There does not appear to be much reason to expect instability in the manipulation of number series. However, since errors will build up faster than with the analytical method, the point may be reached sooner where round-off errors will cause wide fluctuations in the outputs. Hence, this method is rated second in this characteristic rather than first.

This method stands last by a significant amount with regard to storage requirements. This is largely due to the fact that no advantage was taken of zero elements in the transition matrix. Gains can be made by a new program which does take such an advantage; however, these gains may be more than offset if shorter time intervals are taken for greater accuracy since this would require longer number series.

### 5.4.5 GENERAL

From the above discussions, it is evident that no one method can be selected as the best for all portions of the LVCLS implementation. The relative advantages will change from one portion to another and can be controlled to a large extent by the trade-offs that can be made. For each linear portion of the system, each method must be considered before the best can be chosen. The most likely result will be that a combination of all four methods will be used, even within subsystems.

For piecewise linear systems, the situation is not much different. The numerical integration method has the advantage that it can handle these cases with the same facility as for linear cases except for a possible increase in running time. When the discontinuties in the system occur as a function of time only, the number series method also has this advantage, and there probably will not be an increase in computer running time. For all other piecewise linear cases, all four methods may be used with proper matching of terminal and initial conditions at places of discontinuties. If there are many regions, the necessity of solving sets of equations several times, and of making these matchings may increase the computer running time of the two transfer function methods and the number series method more than that of the numerical integration method, making this one the best.

For nonlinear portions of the system, numerical integration is the only method than can be used without making approximations. In some cases these approximations will result in errors that are acceptable, but not in all.

# APPENDIX A

## DIRECT SOLUTION BY NUMERICAL INTEGRATION OF DIFFERENTIAL EQUATIONS

### A.1 GENERAL

The method of numerical integration proposed for this simulation program, described in Reference 1, is specifically designed to handle wide ranges of transients, including step inputs and short duration pulses at any time during a run, and to minimize the propagation and accumulation of errors at all times during a run.

The numerical procedure is presently programmed and running in FORTRAN IV on the IBM 7044 computer. It is scaled to handle up to 20 nonlinear, first-order differential equations. There is no problem involved in expanding the base to meet the requirements of larger system modeling.

### A.2 NUMERICAL INTEGRATION

An integration algorithm using higher derivatives with automatic stabilization has been developed. A higher-derivative formula was selected because of the extreme simplicity obtained with respect to step size control. It has long been known that higher-derivative integration formulas obtain a minimum truncation error with respect to processed information. By adding an additional correction which takes advantage of the derivatives already having been calculated, an automatic numerical stabilization is achieved. The formulas are readily derived using Taylor's series.

The method has been tried on several test cases and has demonstrated excellent tracking capability. By reducing interval size, the method passes through finite discontinuities (such as step changes) with reasonable computational speed. An interpolation formula which uses the derivatives eliminates the necessity of restarting the solution.

Because the truncation error is essentially invariant at any given point, errors are not propagated. This means numerical stability is guaranteed for each accepted point.

The method requires evaluation of first, second, and third derivatives which for nonlinear equations is about equal to four function evaluations with respect to computation time. The procedure for maintaining step size is predict-correct-correct-modify-correct-correct. The predictor represents one integration. The first corrector represents one integration and four function evaluations. The same holds for the second corrector. The modifier involves only four function evaluations. The remaining two correctors require one integration each. Hence, for continuing with constant step size, 12 evaluations and five integrations are required. This does not change if the interval size is doubled. For halving the interval size, an additional integration plus four function evaluations are required (16 evaluations and six integrations). Experience has shown that evaluation of the equations takes about the same time as do the integrations; therefore, each integration by this method requires less than 18 equation evaluations.

In contrast, in order to obtain similar stability control, the four-term Runge Kutta (RK-4) method can be compared. Each step of RK-4 requires four function evaluations plus one integration. Using this approach, however, three integration schemes must be carried on simultaneously in order to obtain optimum computation speed. The doubler can use a previous point and so requires only four evaluations. The half-step scheme requires two integrations, and hence, seven evaluations. This totals 15 evaluations and four integrations. The integration scheme of RK-4 is about twice as fast as an equation evaluation. Hence, this also totals less than 18 equation evaluations.

On a point-to-point basis RK-4 is slightly slower; however, RK-4 does not enjoy most of the other advantages of using higher derivatives. Its main claim to fame is that it is self-starting. It propagates errors and is less accurate. RK-4 has 5th-order accuracy, while the derivative technique has 8th-order accuracy (with respect to first neglected derivative and step size). In order to approximate the truncation error in RK-4, considerable computation is required. One need only compare the results of the last two corrections in the derivative scheme.

There seems to be no great difficulty computing second and third derivatives. In general, these formulas can be written down by inspection. An additional function evaluation was included in the above computation to account for increased complexity of these higher derivatives; however, this seems to be more than is actually required in most cases.

## A.3 NUMERICAL INTEGRATION OF DIFFERENTIAL EQUATIONS

In order to possess an accurate, stable, and reasonably fast integration technique for solving large differential equation systems, the following algorithm had been chosen. This algorithm is of the predictor-corrector type with automatic step size modification. The integrator applies the following formulas:

$$
\begin{aligned}
\text{P:}\quad y_p(x+h) \;=\;& y(x-h) + 2h\left[4\frac{d}{dx}y(x) - 3\frac{d}{dx}y(x-h)\right] \\
& - \frac{2}{5}h^2\left[8\frac{d^2}{dx^2}y(x) + 7\frac{d^2}{dx^2}y(x-h)\right] \\
& + \frac{2}{15}h^3\left[7\frac{d^3}{dx^3}y(x) - 3\frac{d^3}{dx^3}y(x-h)\right]
\end{aligned}
\qquad \text{(A-1)}
$$

where $y(x - h)$ and $y(x)$ have previously been computed together with their corresponding first, second, and third derivatives.

If $y_T(x + h)$ is the true value of y at $x + h$, then the mean value truncation error is

$$
y_T(x+h) = y_p(x+h) + \frac{208}{100800}h^7\frac{d^7}{dx^7}y(x+\theta h), \qquad \text{(A-2)}
$$

$(0 < \theta < 1)$; h is the step size.

$$
\begin{aligned}
\text{C:}\quad y_c(x+h) \;=\;& y(x) + \frac{1}{2}h\left[\frac{d}{dx}y_p(x+h) + \frac{d}{dx}y(x)\right] \\
& - \frac{1}{10}h^2\left[\frac{d^2}{dx^2}y_p(x+h) - \frac{d^2}{dx^2}y(x)\right] \\
& - \frac{1}{120}h^3\left[\frac{d^3}{dx^3}y_p(x+h) + \frac{d^3}{dx^3}y(x)\right]
\end{aligned}
\qquad \text{(A-3)}
$$

where $y(x)$ and its derivatives have previously been computed; [$y_p(x + h)$ is the value obtained from the predictor]. This corrector is applied twice; first using derivatives computed from predicted values and second using derivatives computed from the first correction. Let these be $y_{c1}(x + h)$ and $y_{c2}(x + h)$, respectively. Then,

$$
y_T(x+h) = y_{c1}(x+h) - \frac{1}{100800}h^7\frac{d^7}{dx^7}y(x+\theta h) + \frac{104}{100800}h^8\sum\frac{d^7}{dx^7}z(x+\theta h)\frac{\partial f}{\partial z}f(\zeta) \qquad \text{(A-4)}
$$

where z ranges over all of the variables of the system of equations and $f(z) = dz/dx$ with $\xi$ having its usual mean value definition.

For the second correction:

$$y_T(x+h) = y_{c2}(x+h) - \frac{1}{100800} h^7 \frac{d^7}{dx^7} y(x+\theta h)$$

$$- \frac{1}{201600} h^8 \sum \frac{d^7}{dx^7} z(x+\theta h) \frac{\partial}{\partial z} f(\xi) \qquad \text{(A-5)}$$

Further iteration of the corrector does not reduce the mean value error:

$$\text{M:} \quad y_M(x+h) = y_{c2}(x+h) + \frac{1}{209}\left[y_p(x+h) - y_{c2}(x+h)\right] \qquad \text{(A-6)}$$

For this:

$$y_T(x+h) = y_M(x+h) - \left(\frac{104}{209}\right) \frac{1}{100800} h^8 \sum \frac{d^7}{dx^7} z(x+\theta h) \frac{\partial}{\partial z} f(\xi)$$

The final basic integration formula is used for step size reduction

$$\text{R:} \quad y_R(x) = \frac{1}{2}\left[y(x+h) + y(x-h)\right]$$

$$- \frac{11}{32} h\left[\frac{dy}{dx}(x+h) - \frac{dy}{dx}(x-h)\right]$$

$$+ \frac{3}{32} h^2\left[\frac{d^2}{dx^2} y(x+h) + \frac{d^2}{dx^2} y(x-h)\right] \qquad \text{(A-7)}$$

$$- \frac{1}{96} h^3\left[\frac{d^3}{dx^3} y(x+h) - \frac{d^3}{dx^3} y(x-h)\right]$$

$$y_T(x) = y_R(x) + \frac{1}{2240} h^8 \frac{d^8}{dx^8} y(x+\theta h)$$

$$(-1 < \theta < +1).$$

All of the above formulas and mean value error terms are derivable from Taylor's series. Although the derivation assumes the existence of at least eight derivatives, the algorithm seems to work fairly well when faced with finite discontinuities.

As indicated above, the basic integration scheme is predict-correct-correct-modify. After computing a set of modified values, the corrector is used again, and the difference between these new corrected values and the modified values is required to be small to insure numerical stability. If this difference is not small, R is used to compute values at $x - \frac{1}{2}h$, and this new point is used to continue the integration at half the step size.

If this difference is small, the corrector is again applied between values at $x - h$ and $x + h$ (step size 2h). These new corrected values are compared with the modified values. If this difference is small, the step size is doubled and $y(x - h)$ and $y(x + h)$ are used to continue at twice the step size. If it is not small, the present step size is retained. The following flow chart (Figure A-1) indicates the essential features of the algorithm.

## A.4 RESULTS OF TEST CASES

Three specific kinds of nonlinear system equations have been programmed and run using this method of numerical integration. They are:

- Three-loop nonlinear hydraulics model
- Gyro and servo loop with torquer output limit
- Nuclear reactor kinetics equations.

The results of the first two cases are shown in Figures A-2 through A-8. The results of the last case are given in Reference 1.

### A.4.1 THREE-LOOP NONLINEAR HYDRAULICS MODEL

Figure A-2 shows the assumed hydraulic model which was used as the original test case. Details of the model and results are covered in Reference 3, which reported an analog computer study of the same system. It is important to note that this was a comparative study of the same problem by two entirely different approaches which gave <u>almost</u> precisely the same results. Usage of the term "almost" is qualified by virtue of the fact that the analog was not "tuned up" quite to the point that it could have been. Comparison of the results showed that the analog was slightly in error.

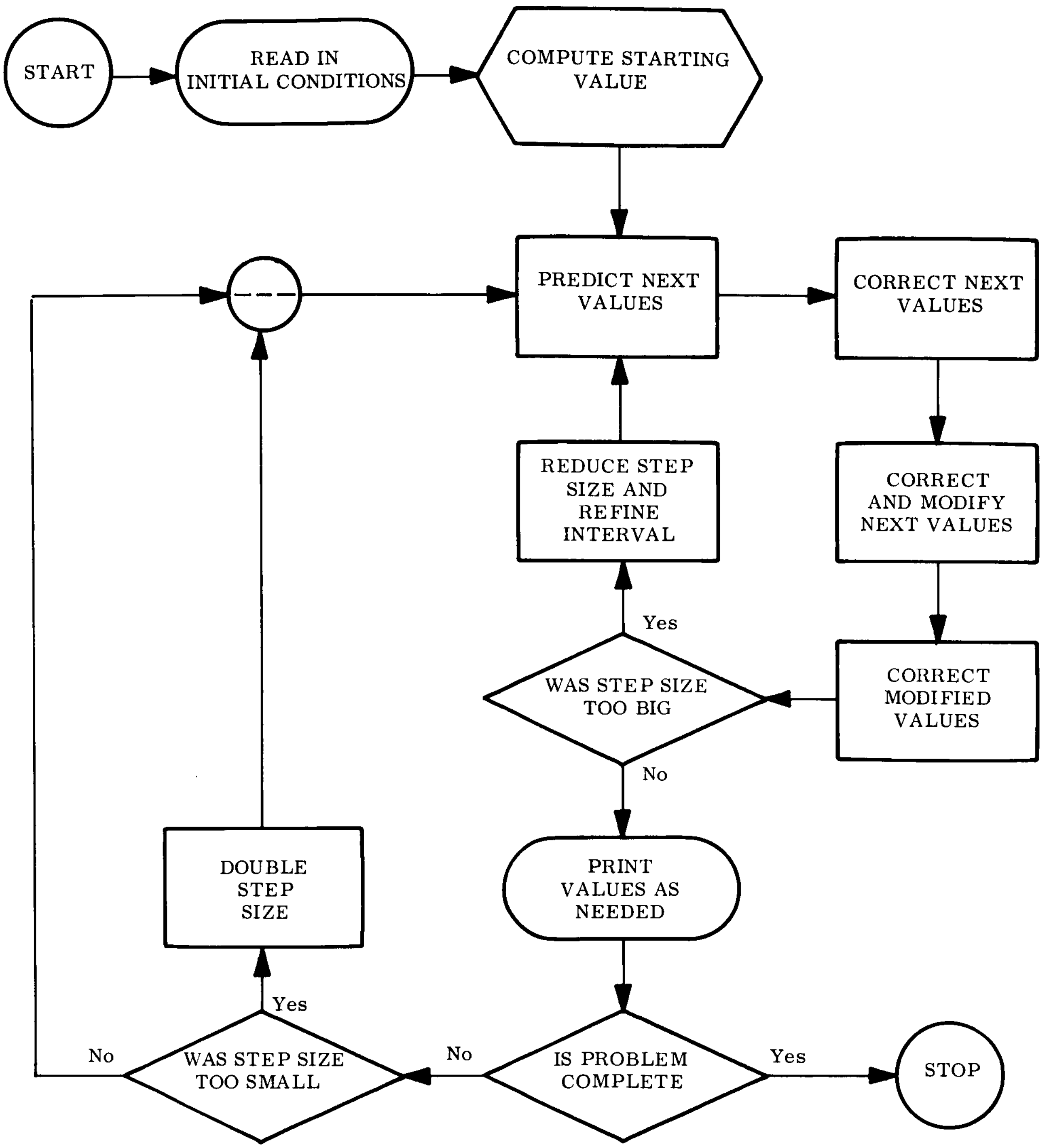



Figure A-1. Flow Chart for EQUSOL

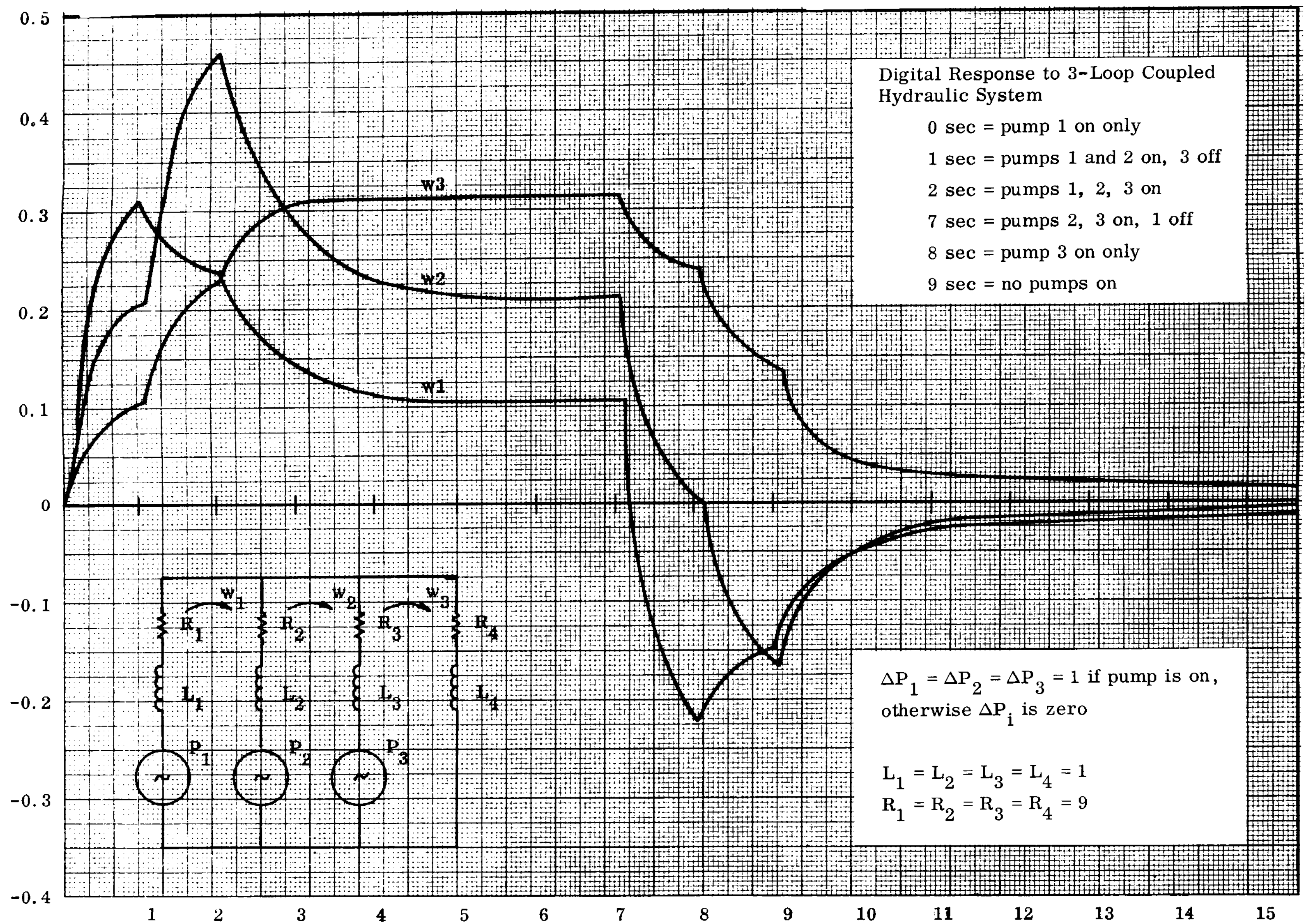



Figure A-2. Digital Response to 3-Loop Coupled Hydraulic System

It is significant to note that the digital model was well able to accept step changes in pump heads at arbitrarily selected times and compute responses without carrying any apparent error and without generating any (numerical) instabilities (such as ringing).

## A.5 DERIVATION OF X-LOOP GYRO EQUATIONS

Figure A-3 represents the network portion of the gyro tables.

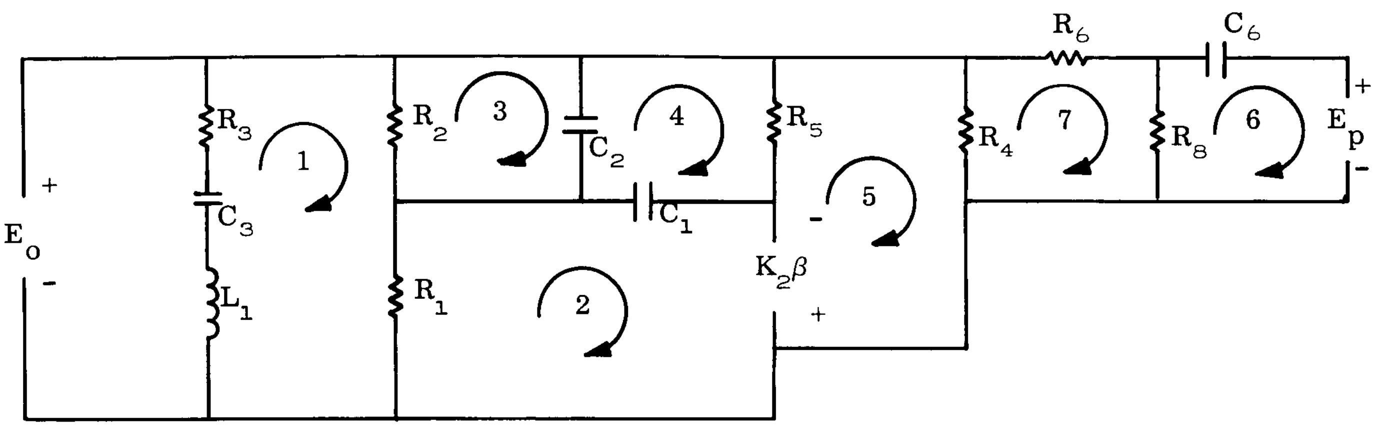



Figure A-3. Gyro System Network

Between the network output voltage $E_o$ and $K_2\beta$, the following two differential equations exist:

For the torquer

$$\frac{d^2E_{tq}}{dt^2} = \omega_8^{\,2}\left(E_o - E_{tq} + \frac{1}{\omega_7}\frac{dE_o}{dt}\right) - 2D_8\,\omega_8\,\frac{dE_{tq}}{dt} \tag{A-8}$$

For the table dynamics

$$\frac{d^3\beta}{dt^3} = \omega^2\left(\frac{K_1E_{tq}}{H} - \frac{d\beta}{dt}\right) \tag{A-9}$$

The network loop equations are:

$$\frac{d^2q_1}{dt^2} = \frac{1}{L_1}\left[-\,(R_1 + R_2 + R_3)\frac{dq_1}{dt} + R_1\frac{dq_2}{dt} + R_2\frac{dq_3}{dt} - \frac{1}{C_3}\,q_1\right] \tag{A-10}$$

$$\frac{dq_2}{dt} = \frac{1}{R_1}\left(R_1 \frac{dq_1}{dt} - \frac{1}{C_1} q_2 + \frac{1}{C_1} q_4 + K_2\beta\right) \qquad \text{(A-11)}$$

$$\frac{dq_3}{dt} = \frac{1}{R_2}\left(R_2 \frac{dq_1}{dt} - \frac{1}{C_2} q_3 + \frac{1}{C_2} q_4\right) \qquad \text{(A-12)}$$

$$\frac{dq_4}{dt} = \frac{1}{R_5}\left[\frac{1}{C_1} q_2 + \frac{1}{C_2} q_3 - \left(\frac{1}{C_1} + \frac{1}{C_2}\right) q_4 + R_5 i_5\right] \qquad \text{(A-13)}$$

$$i_5 = \frac{1}{R_4 + R_5}\left(R_5 \frac{dq_4}{dt} + R_4 i_7 - K_2\beta\right) \qquad \text{(A-14)}$$

$$\frac{dq_6}{dt} = \frac{1}{R_8}\left(-\frac{1}{C_6} q_6 + R_8 i_7 - E_p\right) \qquad \text{(A-15)}$$

$$i_7 = \frac{1}{R_4 + R_6 + R_8}\left(R_4 i_5 + R_8 \frac{dq_6}{dt}\right) \qquad \text{(A-16)}$$

Removal of the algebraic loop (A-11 through A-14) led to the following set of equations which were programmed for the IBM 7044 computer:

$$\phi_1 = \left(\frac{1}{C_1} + \frac{1}{C_2}\right) q_4 - \frac{1}{C_1} q_2 - \frac{1}{C_2} q_3 \qquad \text{(A-17)}$$

$$E_o = -(K_2\beta + \phi_1) \qquad \text{(A-18)}$$

$$\frac{d^3\beta}{dt^3} = \omega^2\left(\frac{K_1}{H} E_{tq} - \frac{d\beta}{dt}\right) \qquad \text{(A-19)}$$

$$i_7 = \frac{1}{R_6}\left[E_o - \left(E_p + \frac{1}{C_6} q_6\right)\right] \qquad \text{(A-20)}$$

$$\frac{dq_6}{dt} = i_7 - \frac{1}{R_8}\left(E_p + \frac{1}{C_6} q_6\right) \qquad \text{(A-21)}$$

$$i_5 = i_7 + \frac{1}{R_4} E_o \tag{A-22}$$

$$\frac{dq_4}{dt} = i_5 - \frac{1}{R_5} \phi_1 \tag{A-23}$$

$$\phi_2 = - K_2\beta - \frac{1}{C_1}(q_4 - q_2) \tag{A-24}$$

$$\frac{dq_2}{dt} = \frac{dq_1}{dt} - \frac{1}{R_1}\phi_2 \tag{A-25}$$

$$\frac{dq_3}{dt} = \frac{dq_1}{dt} + \frac{1}{R_2 C_2}(q_4 - q_3) \tag{A-26}$$

$$\frac{d^2q_1}{dt^2} = \frac{1}{L_1}\left\{R_2 \frac{dq_3}{dt} - \left[\phi_2 + (R_2 + R_3)\frac{dq_1}{dt} + \frac{1}{C_3} q_1\right]\right\} \tag{A-27}$$

$$\frac{d^2E_{tq}}{dt^2} = \omega_8^{\;2}\left(E_o - E_{tq} + \frac{1}{\omega_7}\frac{dE_o}{dt}\right) - 2D_8\,\omega_8 \frac{dE_{tq}}{dt} \tag{A-28}$$

$q_j$ in every case above equals $\int_o^t i_j dt$.

The limiter was inserted by limiting the value of $E_{tq}$ in Equation A-19.

Runs have been made for a step input of 1 volt (at $E_p$), which did not actuate the limiter; a 10-volt step input, which did actuate the limiter; and a 10-volt, 12 millisecond pulse which actuated the limiter. Problem time for the step was 400 milliseconds. Using a tight error criterion of $10^{-7}$ for step control, this required about 20 minutes of computation time giving a ratio of 0.4/1200 = 1/3000. For the pulse input, 400 milliseconds required 12 minutes or a ratio of 0.4/720 = 1/1800. The increased speed is very likely due to the smoother response after the pulse was removed.

Figures A-4 through A-8 are graphs related to deflection angle, $\beta$, versus time for these runs.

Figure A-4 is $\beta$ versus time for a 1-volt step input evaluated from the transfer function response.

Figure A-5 compares $\beta$ versus t for a 10-volt step with and without the torquer limit of 0.10741 volt.

The no-limit curve of Figure A-5 is a direct overlay of the curve of Figure A-4. This is excellent agreement.

Figures A-6 through A-8 are, respectively,

$$\beta, \frac{d\beta}{dt}, \frac{d^2\beta}{dt^2}, \frac{d^3\beta}{dt^3} \text{ versus t for the pulse input.}$$

These curves were plotted for the purpose of checking the limiter logic of the differential equation evaluator. The extreme points of each successive derivative occur (well within plotting accuracy) at the same values of t as the zeros of the corresponding next derivative. This represents an excellent check upon the numerical stability of the procedure.

## A.6 EFFECT OF CHANGING ACCURACY CRITERION

In applying the numerical integration method to the gyro problem, an accuracy criterion of $10^{-7}$ was chosen for the step size control. This is much greater accuracy than needed in a practical problem, but this represents the possible error in each integration step, and it was not known how errors would build up with a large number of steps. This figure was chosen in hopes that overall errors would not become more than $10^{-4}$.

As a typical run, the step input without the limiter was chosen, and errors were calculated for selected values of time. These values lay in two ones on the increasing slope of $\beta$, one around the maximum, and two zones on the decreasing slope of $\beta$. Surprisingly, the errors were much less than expected. Instead of $10^{-4}$, they were actually less than $10^{-7}$ for the points calculated.

As a further check on errors and on computer running time, as influenced by the error criterion, additional runs were made with everything the same but error criterion. These were chosen to be $10^{-5}$ and $10^{-3}$. Errors were calculated for the same time points. The results of all three calculations are shown in Table A-1.

Table A-1

Results of Runs With Various Error Criteria

| t | $\beta_e$ | $\beta_{i_3}$ | $\beta_{i_5}$ | $\beta_{i_7}$ | $10^8 \Delta\beta_3$ | $10^8 \Delta\beta_5$ | $10^8 \Delta\beta_7$ |
|---|---|---|---|---|---|---|---|
| 19 | 0.61792499 | 0.61792486 | 0.61792519 | 0.61792506 | -13 | 20 | 7 |
| 20 | 0.66814967 | 0.66814939 | 0.66814986 | 0.66814973 | -28 | 19 | 7 |
| 21 | 0.71836857 | 0.71836817 | 0.71836876 | 0.71836863 | -40 | 19 | 6 |
| 34 | 1.27087050 | 1.27087051 | 1.27087064 | 1.27087056 | 1 | 14 | 6 |
| 35 | 1.30088823 | 1.30088814 | 1.30088836 | 1.30088829 | - 9 | 13 | 6 |
| 36 | 1.32915332 | 1.32915314 | 1.32915345 | 1.32915337 | -18 | 13 | 5 |
| 49 | 1.53287440 | 1.53287504 | 1.53287445 | 1.53287443 | 64 | 5 | 3 |
| 50 | 1.53671900 | 1.53671955 | 1.53671904 | 1.53671902 | 55 | 4 | 2 |
| 51 | 1.53910643 | 1.53910695 | 1.53910649 | 1.53910645 | 48 | 6 | 2 |
| 52 | 1.54007469 | 1.54007513 | 1.54007474 | 1.54007470 | 44 | 5 | 1 |
| 53 | 1.53965917 | 1.53965960 | 1.53965922 | 1.53965920 | 43 | 5 | 3 |
| 54 | 1.53789503 | 1.53789547 | 1.53789507 | 1.53789504 | 44 | 4 | 1 |
| 79 | 1.19018001 | 1.19018035 | 1.19017997 | 1.19018002 | 34 | 1 | 1 |
| 80 | 1.16938816 | 1.16938847 | 1.16938814 | 1.16938817 | 31 | -2 | 1 |
| 81 | 1.14842376 | 1.14842405 | 1.14842372 | 1.14842377 | 29 | -4 | 1 |
| 109 | 0.59280369 | 0.59280353 | 0.59280366 | 0.59280370 | -16 | -3 | 1 |
| 110 | 0.57650777 | 0.57650762 | 0.57650773 | 0.57650778 | -15 | -4 | 1 |
| 111 | 0.56054880 | 0.56054862 | 0.56054877 | 0.56054882 | -18 | -3 | 2 |
| Computer running time for 200 milliseconds problem time; minutes: | | | | | 2.75 | 5.25 | 13 |

t = time in milliseconds

$\beta_e$ = angle in milliradians by direct evaluation

$\beta_{i_3}$ = angle in milliradians by numerical integration with error criterion of $10^{-3}$

$\beta_{i_5}$ = angle in milliradians by numerical integration with error criterion of $10^{-5}$

$\beta_{i_7}$ = angle in milliradians by numerical integration with error criterion of $10^{-7}$

$\Delta\beta_3 = \beta_{i_3} - \beta_e$

$\Delta\beta_5 = \beta_{i_5} - \beta_e$

$\Delta\beta_7 = \beta_{i_7} - \beta_e$

As expected, the running times decreased appreciably with the decreased error criterion, while the errors increased. But the errors were still less than $10^{-6}$, even with $10^{-3}$ as an error criterion.

The explanation for this apparent impossibility was found by examining the computer printout sheets. The error criterion is applied at every integration step to each variable of the set of equations. It was found that some of these variables were much larger in magnitude than $\beta$ and controlled the step size and overall accuracy. Hence, errors in the fourth decimal place were actually errors in the seventh or eighth significant figure, and this determined the accuracy of the output, $\beta$.

In most practical cases, even greater errors could be tolerated than have been found for the $10^{-3}$ run. However, for this gyro problem, it appears that the point of diminishing returns may have been reached. At $10^{-3}$ for a criterion, the errors are starting to build up faster than running time goes down, so that not much would be gained in time by accepting greater errors.

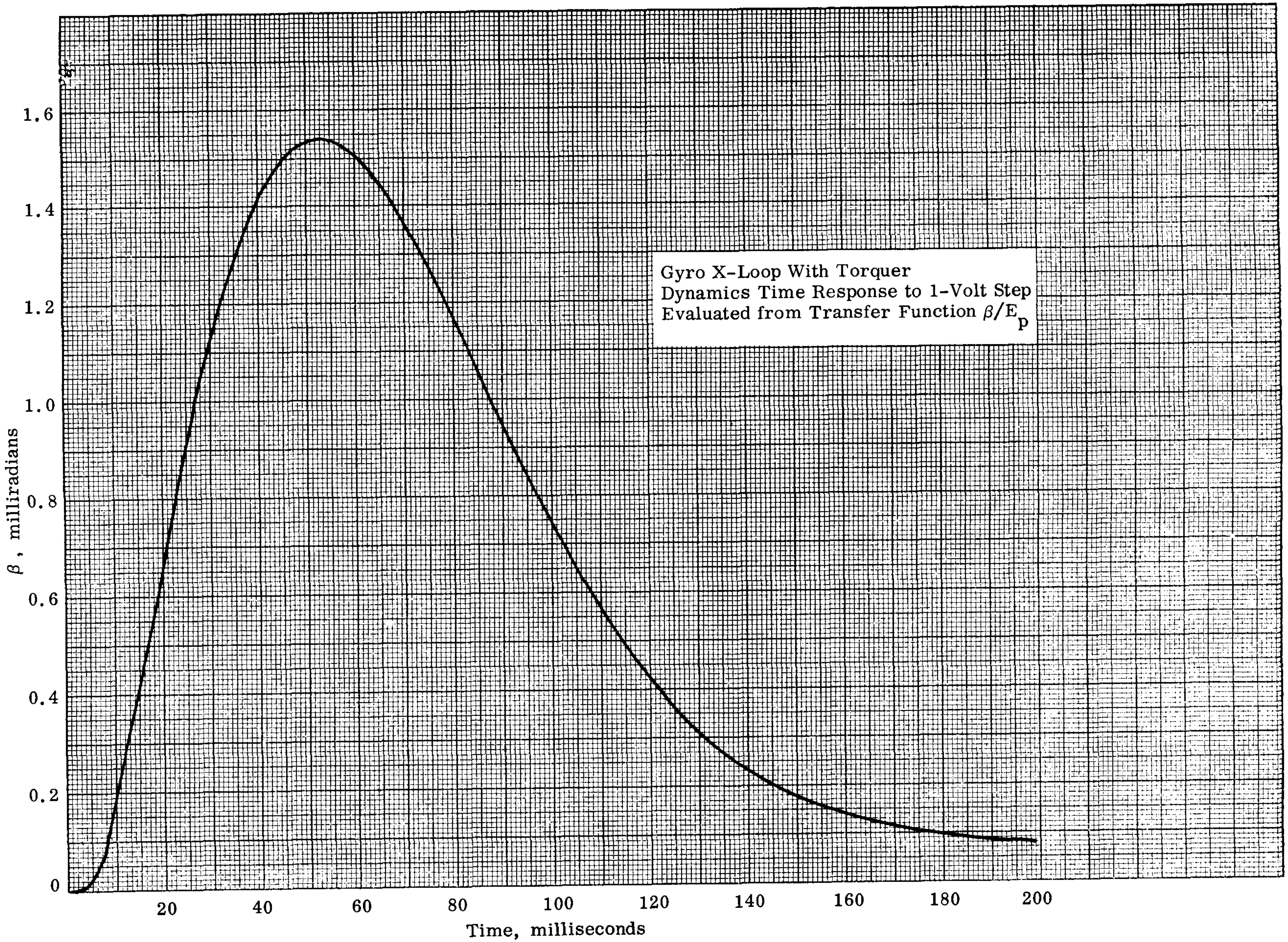
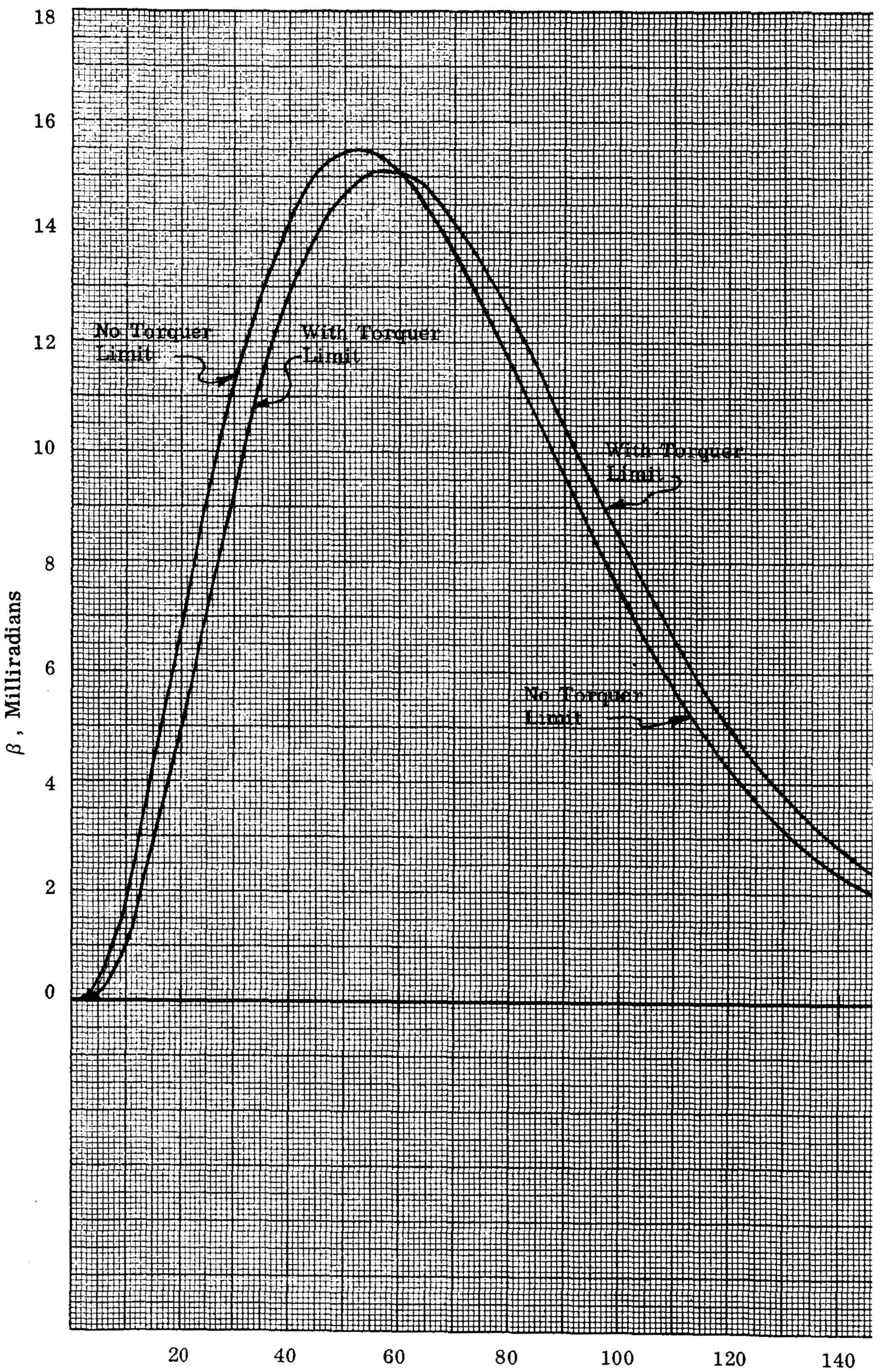



Figure A-4. Gyro X-Loop With Torquer






FOLDOUT FRAME

2

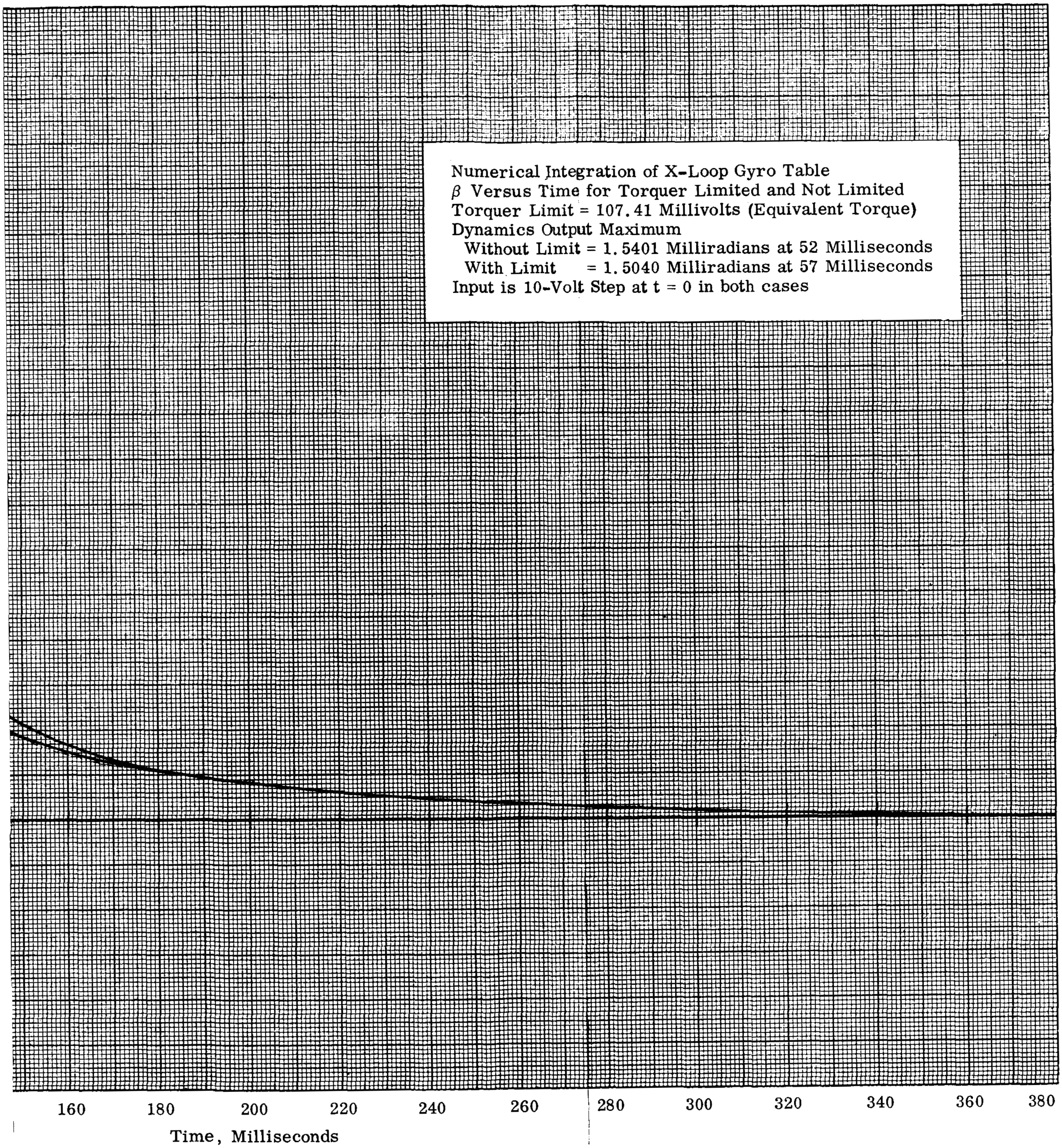
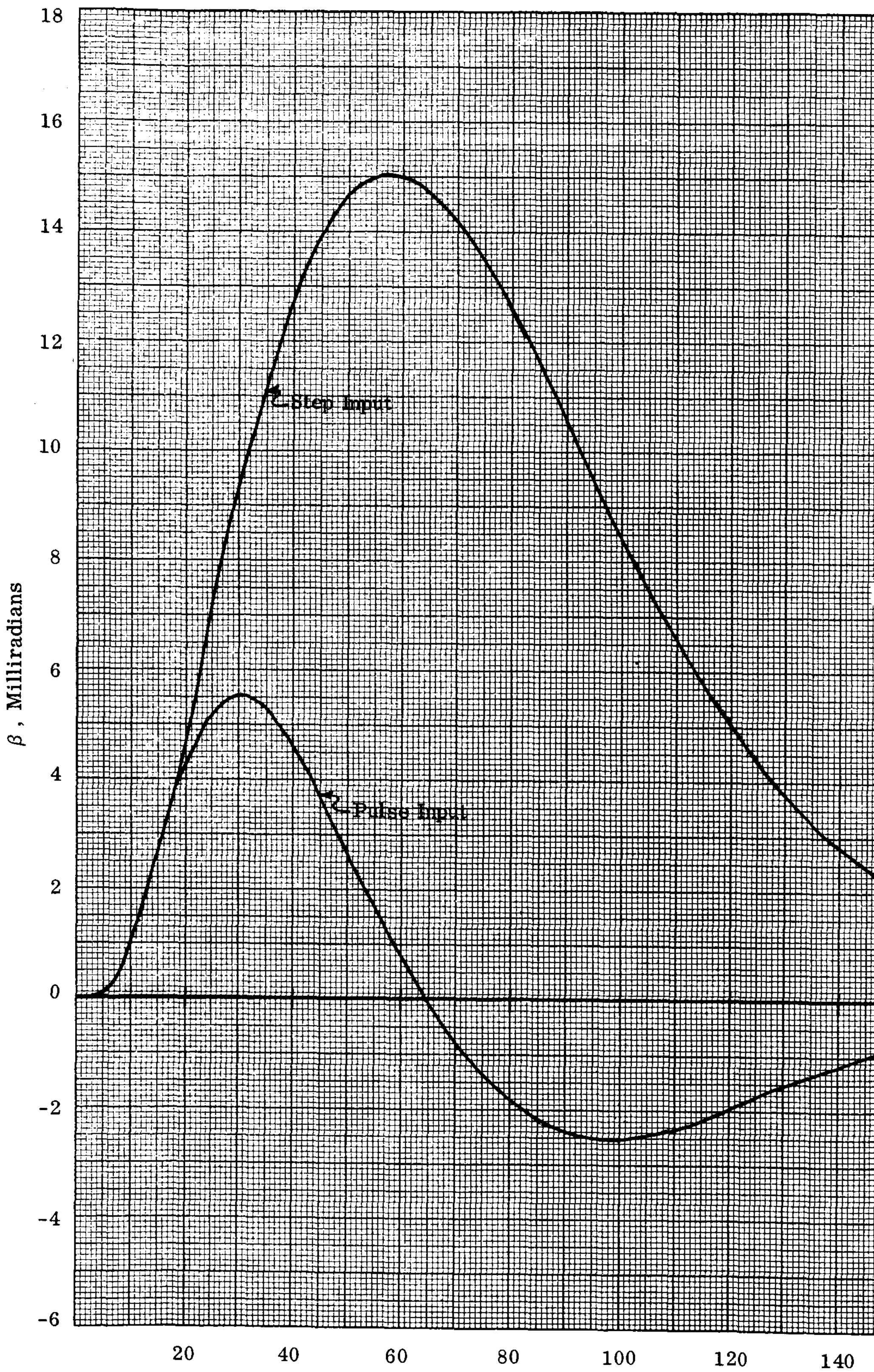



Figure A-5. Numerical Integration of X-Loop Gyro Table $\beta$ Versus Time for Torquer Limited and Not Limited

FOLDOUT FRAME

/






FOLDOUT FRAME

2

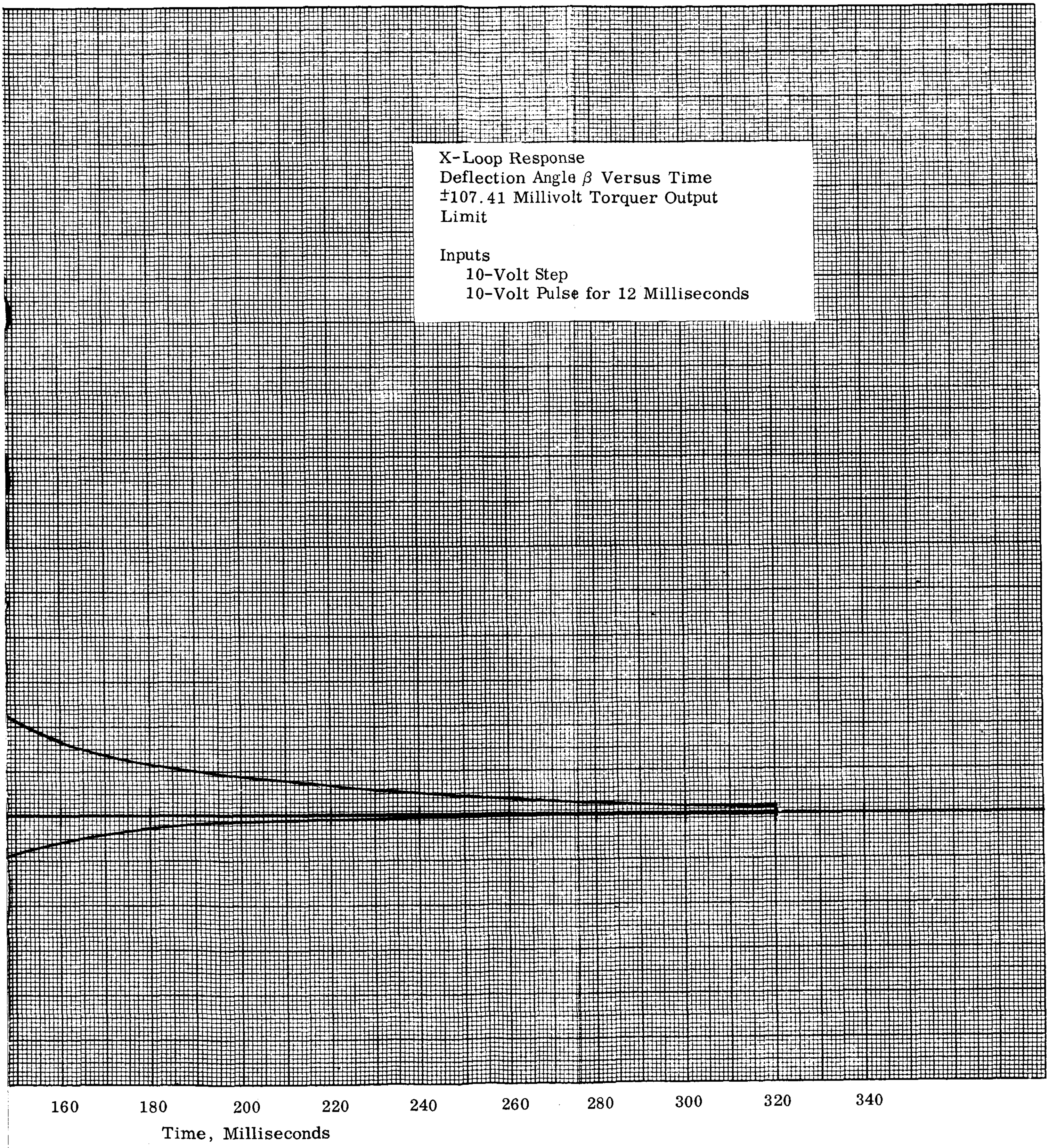
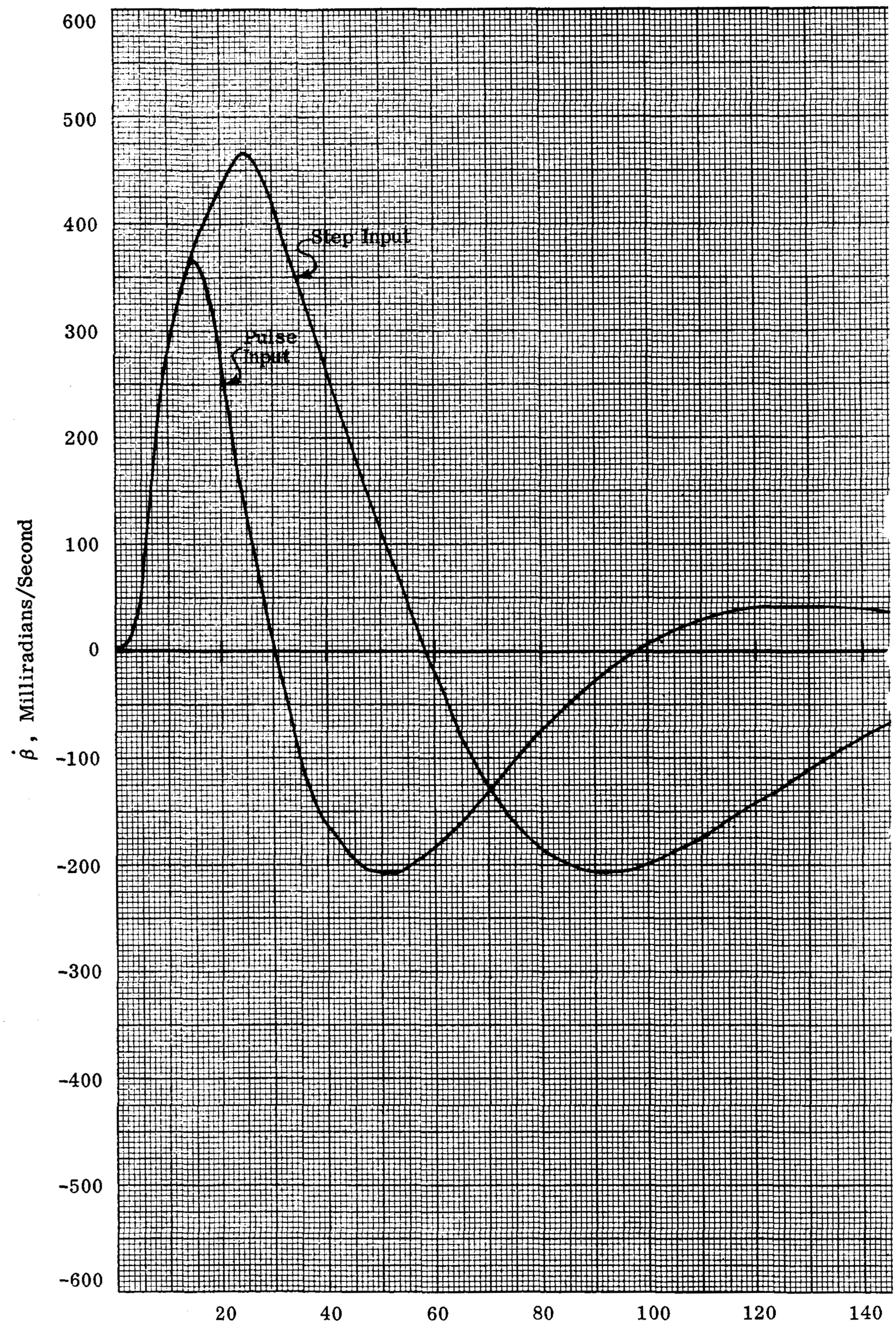



Figure A-6. X-Loop Response Deflection Angle $\beta$ Versus Time




A-19

FOLDOUT FRAME

2

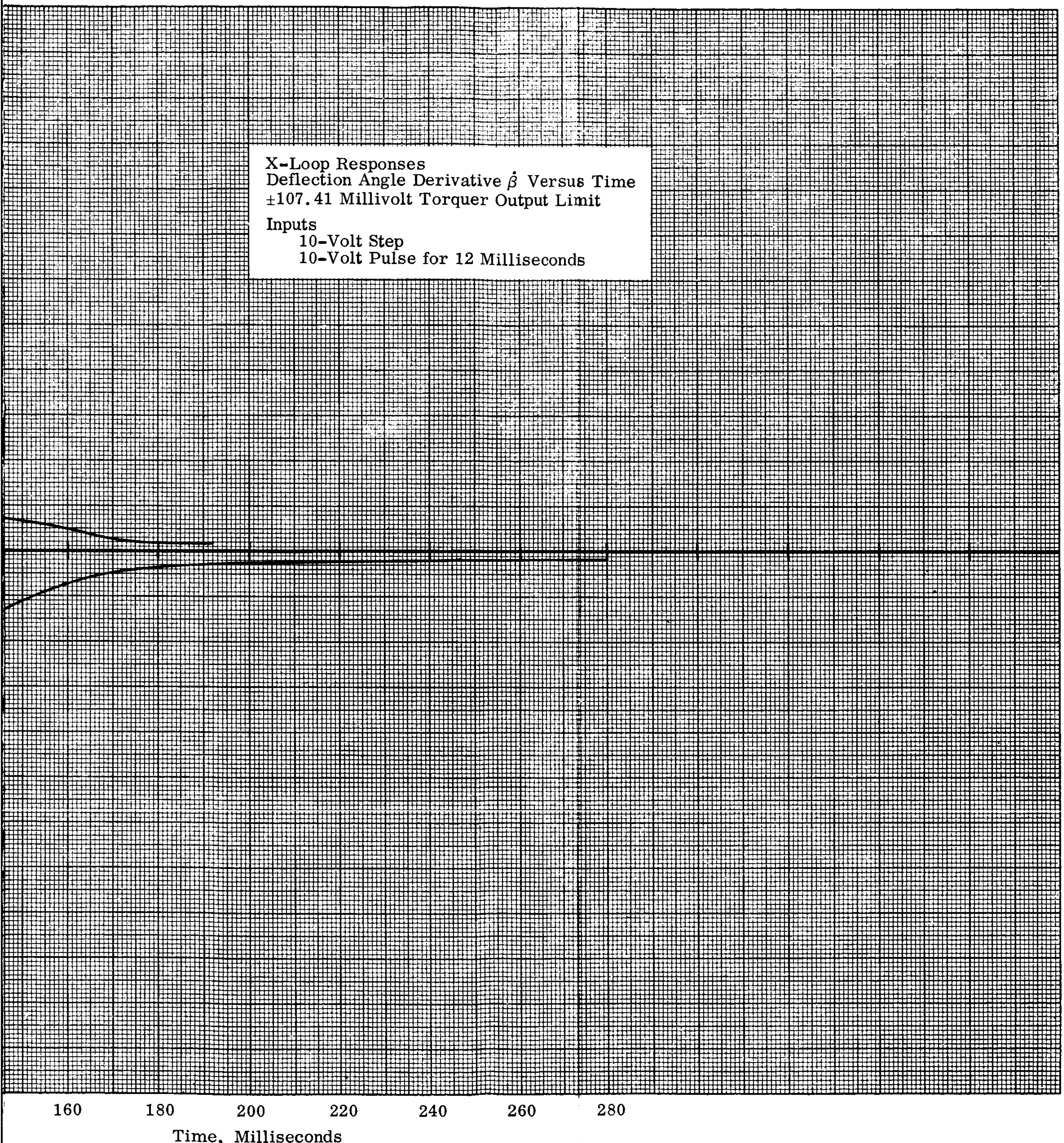



Figure A-7. X-Loop Responses Deflection Angle Derivative $\dot{\beta}$ Versus Time

/

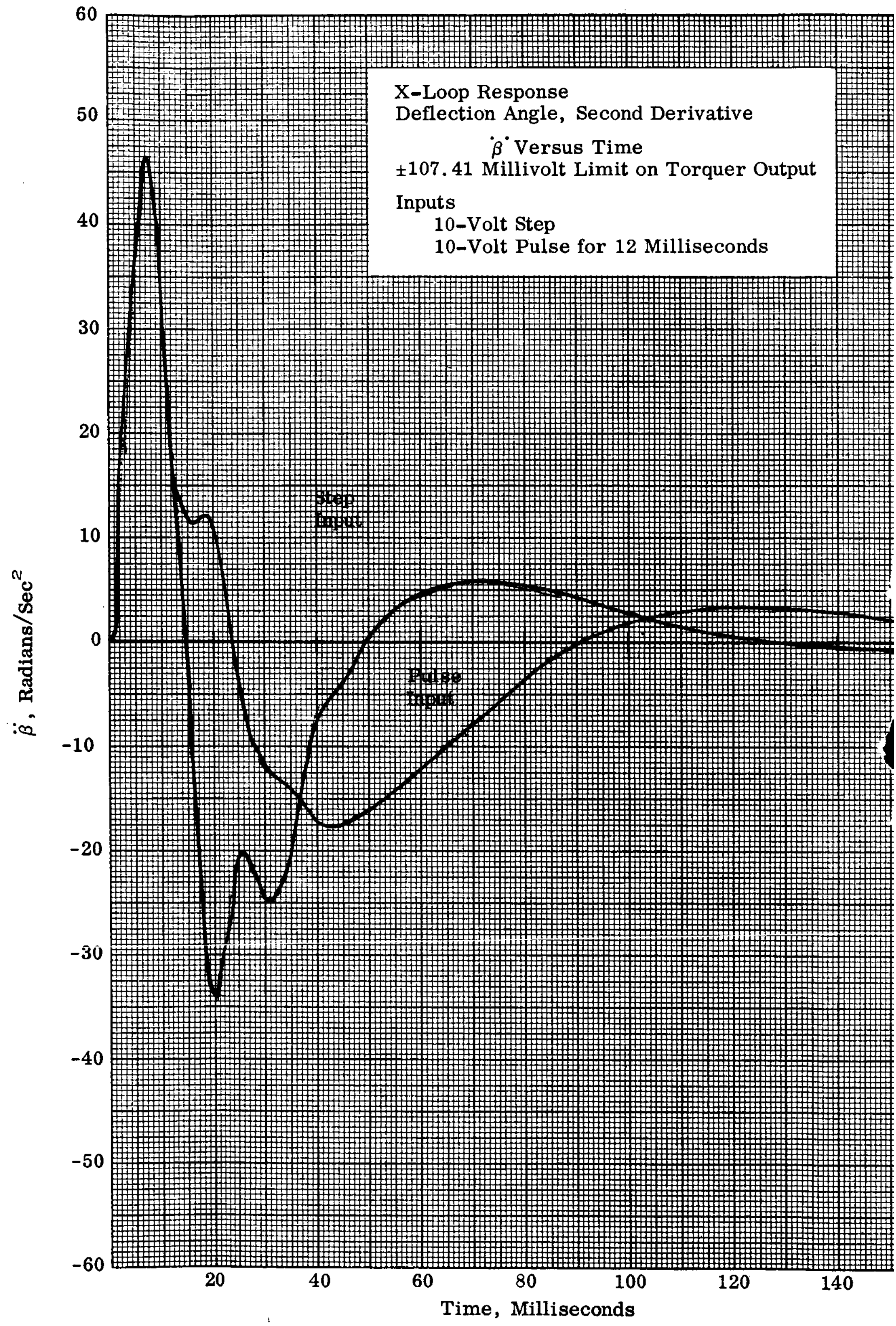
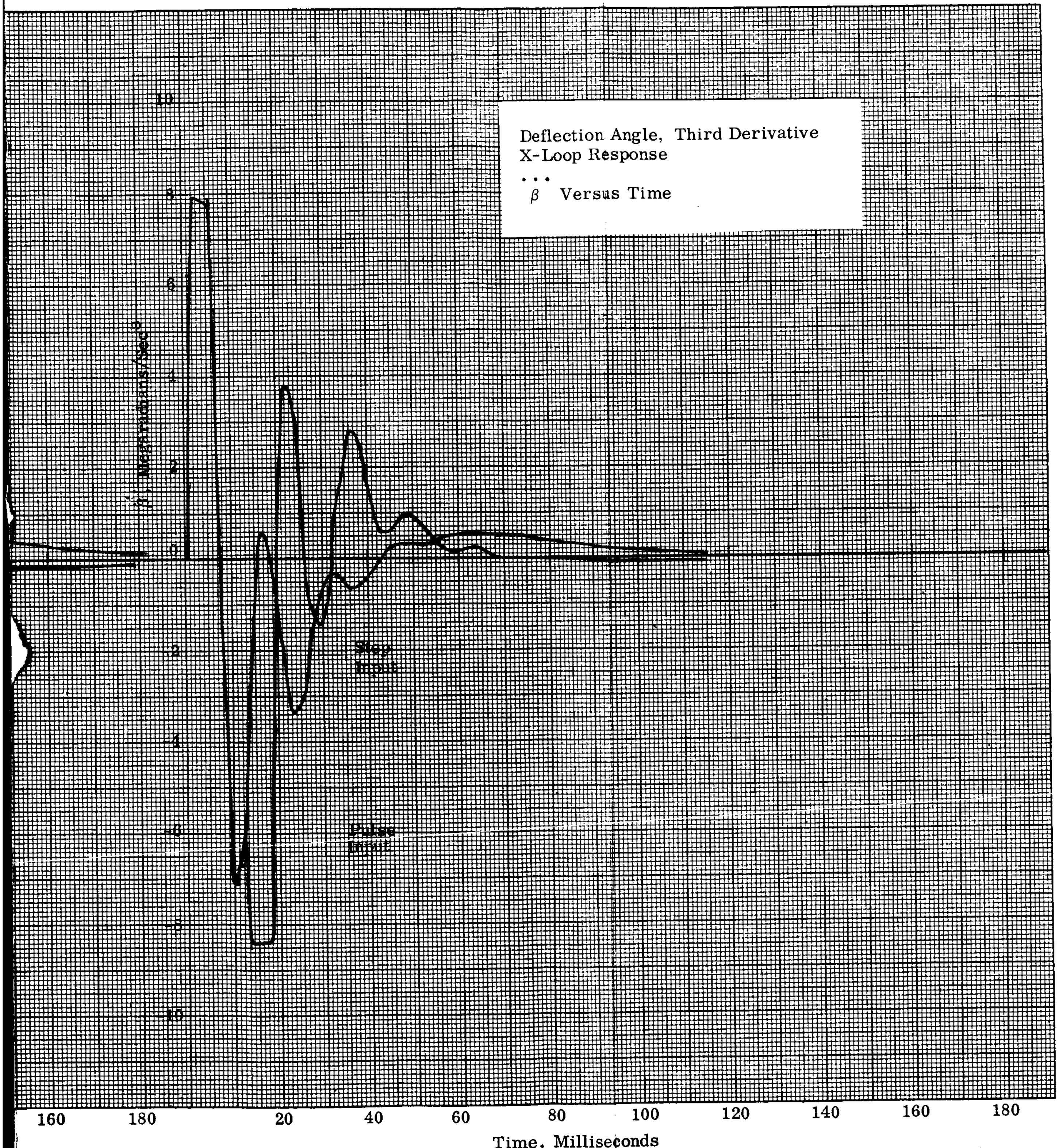



Figure A-8. X-Loop Response Deflection Angle, Second Derivative $\ddot{\beta}$ V



FOLDOUT FRAME

2




ersus Time Figure A-9. Deflection Angle, Third Derivative X-Loop Response $\dddot{\beta}$ Versus Time

## APPENDIX B

## ANALYTICAL DETERMINATION OF SYSTEM TRANSFER FUNCTION FROM SUBSYSTEM OR COMPONENT TRANSFER MATRIX

### B.1 INTRODUCTION

The direct integration of the differential equations of a system may be used on both linear and nonlinear systems. When the system is linear, or may be approximated by linear equations in specified regions, other methods are available. One such method is the analytical determination of the system transfer function from the component or subsystem transfer functions.

To determine the transfer function requires the solution of a set of simultaneous algebraic equations in which some or all of the coefficients are transfer functions, or ratios of polynomials in s. In studying ways of solving such equations, it was soon found that any method that entailed dividing one polynomial by another resulted in building up the degrees of the solution polynomials to many times their theoretical values. But by clearing the equations of fractions and by using a method of solution that requires only multiplications and additions, this build-up does not occur. This is the procedure adopted for use on the trial gyro problem.

Specifically, a program (DSOLVE) has been devised for solving sets of equations with polynomial coefficients by Cramer's Rule and for evaluating the determinants required by the fundamental definition of a determinant. The program will handle ten equations with polynomial coefficients of fourth degree or less.

It should be noted that if there is a large number of nonzero terms in such a set of equations, the computer running time may be many hours. However, for the actual gyro problem, the time was less than a minute.

## B.2 DESCRIPTION OF EQUATIONS

The trial gyro problem was as shown in Figures B-1 and B-2.

Using numerical values from Reference 2, the following equations were derived from Figures B-1 and B-2:

$$I_2 - I_3 - I_4 + I_5 = 0 \tag{B-1}$$

$$1000I_2 + 4.7sI_3 + 0.572958s\beta = 0 \tag{B-2}$$

$$190I_2 + 38sI_4 - 47.69sI_5 - 38sI_7 - 38sI_9 - 10^{-3}sE_o = 0 \tag{B-3}$$

$$3.196sI_3 - 1000I_4 - 6.8 \times 10^{-4}sE_o = 0 \tag{B-4}$$

$$1000I_4 + 34.68sI_5 = 0 \tag{B-5}$$

$$(3s^2 + 615s + 10^4) I_7 - 0.3sE_o = 0 \tag{B-6}$$

$$16I_9 + 10I_{10} - 10^{-4}E_o = 0 \tag{B-7}$$

$$-10I_9 + (s + 10) I_{10} = 10^{-5}sE_p \tag{B-8}$$

$$(562.66248s + 3,094,081) E_o - (s^2 + 1196.12s + 3,094,081) E_t = 0 \tag{B-9}$$

$$73,439.776E_t - s(s^2 + 1521)\beta = 0 \tag{B-10}$$

It will be noted that some of the variables shown in Figures B-1 and B-2 have been eliminated from the equation. This was done to fit the problem within the limitations of the program DSOLVE, which had already been made for a set of ten equations, in order to be consistent with other methods of attending this problem.

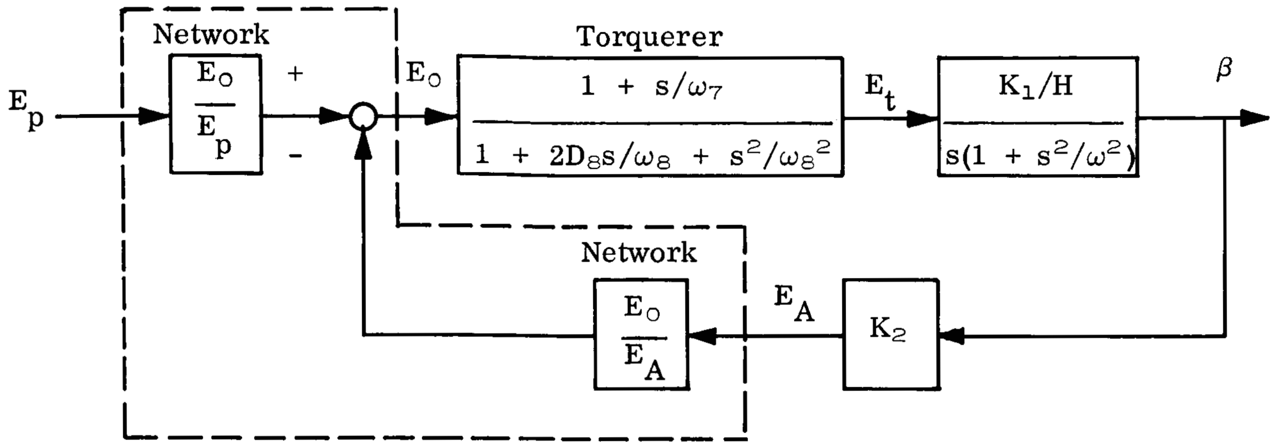
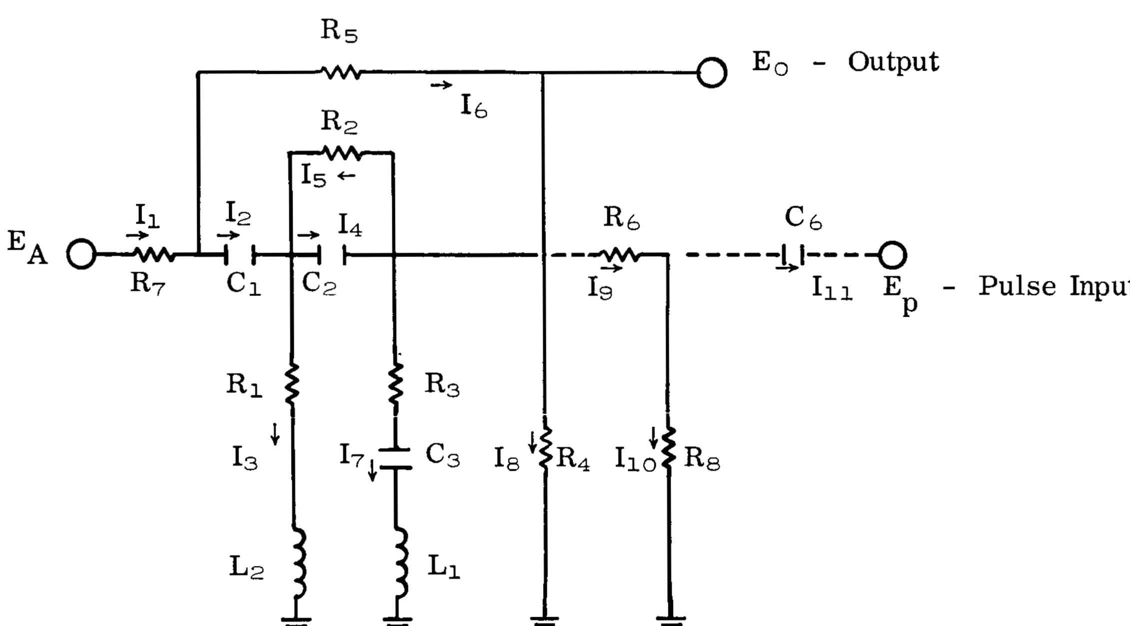



Figure B-1. Gyro System Block Diagram




Note: Circuitry added for test indicated by dashed lines.

Figure B-2. Network Configuration

### B.2.1 RESULTS

Solution of Equations B-1 through B-10 for $\beta$ in terms of $E_p$ gives the following transfer function:

$$\frac{\beta}{E_p} = \frac{N_1}{D_1} \tag{B-11}$$

where

$$N_1 = 6.3805754 \times 10^8 s (s^4 + 5.8423175 \times 10^3 s^3 + 1.9195926 \times 10^6 s^2 + 1.7471594 \times 10^8 s + 2.5353641 \times 10^9)$$

$$D_1 = s^{10} + 1.7517806 \times 10^3 s^9 + 4.1068262 \times 10^6 s^8 + 2.1811184 \times 10^9 s^7 + 1.1563751 \times 10^{12} s^6 + 2.7865710 \times 10^{14} s^5 + 3.8451928 \times 10^{16} s^4 + 2.5076515 \times 10^{18} s^3 + 9.5610742 \times 10^{19} s^2 + 1.7798223 \times 10^{21} s + 1.2049425 \times 10^{22}$$

This compares with the solution on the time-sharing computer:

$$\frac{\beta_e}{E_p} = \frac{N_O}{D_O} \tag{B-12}$$

where

$$N_O = 6.3805731 \times 10^8 s (s^4 + 5.8423177 \times 10^3 s^3 + 1.9195927 \times 10^6 s^2 + 1.7471595 \times 10^8 s + 2.5353642 \times 10^9)$$

$$D_O = s^{10} + 1.7517806 \times 10^3 s^9 + 4.1068264 \times 10^6 s^8 + 2.1811185 \times 10^9 s^7 + 1.1563751 \times 10^{12} s^6 + 2.7865708 \times 10^{14} s^5 + 3.8451922 \times 10^{16} s^4 + 2.5076511 \times 10^{18} s^3 + 9.5610721 \times 10^{19} s^2 + 1.7798219 \times 10^{21} s + 1.2049421 \times 10^{22}$$

The agreement is excellent and well within what might be expected from the accumulation of round-off errors.

## B.3 BACKGROUND FOR CHOICE OF SOLUTION METHOD

Solution by determinant evaluation for a system of n linear equations in n variables with polynomial coefficients of degrees varying from one to four was selected for computer programming in FORTRAN IV for the IBM 7044 instead of a more conventional matrix inversion technique because of the following:

a. Solution by determinant evaluation produces as a result for each variable a ratio of two polynomials of degree no higher than n times the degree of the polynomial coefficient with the maximum degree.

b. The computer program for solution by determinant evaluation and all required computer system routines fit well within the IBM 7044 computer storage limitations (a total of 25733 octal locations are used).

c. Solution by determinant evaluation requires no storage on or retrieval of intermediate results from external media such as magnetic tapes or discs.

Solution by a matrix inversion technique (Crout's elimination) could not be used because of the following:

a. Crout's elimination produces, as a result for each variable, a ratio of two polynomials of degree possibly as high as $2^{n-1}$ times the degree of the polynomial coefficient with the maximum degree. Because division is required in the matrix inversion process and because division can only be indicated by fractional representation of each polynomial for each step in the inversion process, the degree of the resulting polynomial cascades to the result described above. The determinant evaluation technique requires no division processes and produces no cascading of the resulting polynomial degree beyond n times the degree of the maximum polynomial coefficient.

   Inversion by Crout's method of a tenth-order matrix with polynomial coefficients of degree four could produce solution polynomials of degree 2048; whereas, solution by determinants could produce solution polynomials of degree no higher than 40.

b. Solution by Crout's method requires storage on and retrieval of intermediate results from external media such as magnetic tapes or discs because the IBM 7044 core capacity is not adequate for the storage required in the evaluation of a tenth-order matrix with fourth-degree polynomial coefficients.

However, solution by determinant evaluation is not practical from a computer timing standpoint when the system of equations to be evaluated is not judiciously ordered and contains no zero elements (all n variables are present in each equation of the system).

The speed of this method and its practicality increase with an increase in the number of zero elements in a set of n equations in n variables judiciously arranged so that an equation with more zero elements precedes an equation with the next less number of zero elements. As a very rough example, solution for all ten variables in a tenth-order determinant with all nonzero polynomial elements may require approximately 18 hours of computer running time; whereas, a similar solution with only 25 percent of nonzero polynomial elements of varying degree did take approximately 0.7 minutes of computer running time.

## B.4 DESCRIPTION OF METHOD

For each of the n variables to be solved, the program evaluates two nth-order determinants and produces as a solution the ratio of two polynomials. To evaluate each of these determinants, the program sums the signed products of n! permutations of n of its elements, selected so that for each permutation one and only one element comes from any row, and one and only one element comes from any column.

The program determines the sign to be prefixed to each product of n permuted elements by summing the number of inversions of the numbers of the columns from which the elements of a given permutation are taken. An inversion occurs in a given permutation when an element from a larger numbered column precedes an element from a smaller numbered column. If the sum of the number of inversions for a given permutation is odd, the program prefixes with a minus sign the product of the permuted elements; otherwise, for an even number of inversions the program prefixes the product of the permuted elements with a plus sign.

As soon as the program computes each product of a given permutation of n elements and affixes the proper sign to this product, the program adds the product to the sum of all previously computed permuted element products. The program then proceeds to construct another permutation and to compute its associated product.

The program minimizes multiplications by saving each of the n-1 partial products formed in computing the product of n permuted elements. As soon as the program completes the product of n elements in a given permutation, the program backs up through this permutation only to the point that another unique permutation can be formed using as many as possible of the previously permuted elements in combination with those elements changed to yield a new permutation. The program retrieves the partial product at the backup point and uses it in forming the subsequent products for the new permutation. The program further minimizes multiplications by ceasing the multiplication sequence at any point where a zero element is detected. In this case the program searches for the next permutable nonzero element in the given row in which the zero element is encountered and attempts to permute it with all the nonzero elements permuted and multiplied up to this point. If such an element is found, the permutation and multiplication process continues; otherwise, the program continues the backup process until it can construct another unique permutation of nonzero elements or until it exhausts all possibilities for another unique permutation.

### B.4.1 INPUT DESCRIPTION

Input to the program is entered on the standard IBM 80-column card. The required format of the input card deck for each system of equations to be solved is as follows:

<u>Note A</u>

(1) All values are decimal numbers.

(2) All values must be right adjusted in the indicated columns.

<u>Card 1</u>

| Column Numbers | Mode of Entry | Description of Entry |
|---|---|---|
| 1-5 | Integer | n-number of equations in system (must equal number of variables and $1 \le n \le 10$) |
| 6-10 | Integer | Code to indicate variable to be solved for.<br><u>Acceptance Codes are</u>:<br>77 (solve for n variables)<br>1 (solve for 1st variable)<br>2 (solve for 2nd variable)<br>.<br>.<br>.<br>n (solve for nth variable— $1 \le n \le 10$) |
| 11-15 | Integer | Blank or variable code 1-10 |
| 16-20 | Integer | Blank or variable code 1-10 |
| 21-25 | Integer | Blank or variable code 1-10 |
| 26-30 | Integer | Blank or variable code 1-10 |
| 31-35 | Integer | Blank or variable code 1-10 |
| 36-40 | Integer | Blank or variable code 1-10 |
| 41-45 | Integer | Blank or variable code 1-10 |
| 46-50 | Integer | Blank or variable code 1-10 |
| 51-55 | Integer | Blank or variable code 1-10 |

Note B

(1) Only nonzero elements of the system of an equation need be input.

(2) For each nonzero element of the system, all the constant coefficients of each polynomial forming the coefficients of the system element must be input. These polynomial coefficients are to be input in the order of coefficient of zero-degree term first, coefficient of first-degree term second, etc.

(3) For each nonzero element in the system of equations there must be a set of cards in the format described for Card 2, Card 3, and if Card 3 cannot contain all the data required, Card 3A.

Card 2

| Column Numbers | Mode of Entry | Description of Entry |
|---|---|---|
| 1-5 | Integer | i-row number of system element ($1 \leq i \leq n$) |
| 6-10 | Integer | j-column number of system element ($1 \leq j \leq n$) |
| 11-15 | Integer | Degree of polynomial forming coefficient of system element |

Card 3

| Column Numbers | Mode of Entry | Description of Entry |
|---|---|---|
| 1-20 | Floating Point | Blank or coefficient of 0-degree term of polynomial element (i,j) |
| 21-40 | Floating Point | Blank or coefficient of 1st-degree term of polynomial element (i,j) |
| 41-60 | Floating Point | Blank or coefficient of 2nd-degree term of polynomial element (i,j) |
| 61-80 | Floating Point | Blank or coefficient of 3rd-degree term of polynomial element (i,j) |

Card 3A Required only if polynomial element (i,j) is 4th degree.

| Column Numbers | Mode of Entry | Description of Entry |
|---|---|---|
| 1-20 | Floating Point | Coefficient of 4th-degree term of polynomial element (i,j) |

Note C A blank card must follow the last set of cards in format 2, 3, and 3A, to indicate the end of input of all nonzero elements on left side of equal sign in equation system.

Note D

(1) Restrictions (1) and (2) in Note B apply also to elements in the system on the right side of the equal sign in the equation system.

(2) For each nonzero element on the right side of the equal sign in the equation system there must be a set of cards in the format described for Card 4, Card 5, and if Card 5 cannot contain all the data required, Card 5A.

Card 4

| Column Numbers | Mode of Entry | Description of Entry |
|---|---|---|
| 1-5 | Integer | i-row number of system element ($1 \leq i \leq n$) |
| 6-10 | Integer | Degree of polynomial forming coefficient of system element |

Card 5

| Column Numbers | Mode of Entry | Description of Entry |
|---|---|---|
| 1-20 | Floating Point | Blank or coefficient of 0-degree term of polynomial element (i) |
| 21-40 | Floating Point | Blank or coefficient of 1st-degree term of polynomial element (i) |
| 41-60 | Floating Point | Blank or coefficient of 2nd-degree term of polynomial element (i) |
| 61-80 | Floating Point | Blank or coefficient of 3rd-degree term of polynomial element (i) |

Card 5A Required only if polynomial element (i) is 4th degree.

| Column Numbers | Mode of Entry | Description of Entry |
|---|---|---|
| 1-20 | Floating Point | Coefficient of 4th-degree term of polynomial element (i) |

Note E A blank card must follow the last set of cards in format 4, 5, and 5A to indicate the end of input of all nonzero elements on right side of the equal sign in the equation system.

The program will solve as many different equation systems as there are data decks input in the preceding described formats for Cards 1 through 5.

## B.4.2 OUTPUT DESCRIPTION

a. In the system of equations to be solved, the program prints for each row and column element the values the polynomial coefficients input or assumed to be zero, if no values are input for the given row and column.

b. The program also prints for each row element on the right side of the equal sign in the equation system the values of the polynomial coefficients input or assumed to be zero, if no values are input for the given row.

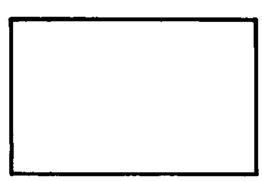
c. For each variable of the system to be solved, the program prints the coefficients of the polynomials forming the numerator and the denominator of the resulting solution. The coefficients are given in the order of increasing degree of the terms of the polynomials.

## B.4.3 FLOW CHART DESCRIPTION

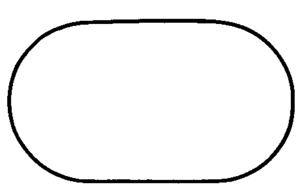
a. In Figures B-3 through B-8, a circled number represents the identical statement number in the FORTRAN IV listing of the program DSOLVE or one of its accompanying subprograms (EVALD, PMULT, or PADD).

b. A circled alphabetic character is used to indicate a connection between steps in the program that are not identified in the FORTRAN IV listing by a statement number.

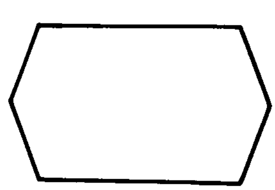
c. The symbols used in the flow chart are described as follows:



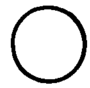
Represents a program step or sequence of steps



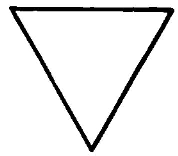
Represents a decision



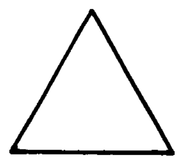
Represents an external subroutine



Represents a connector

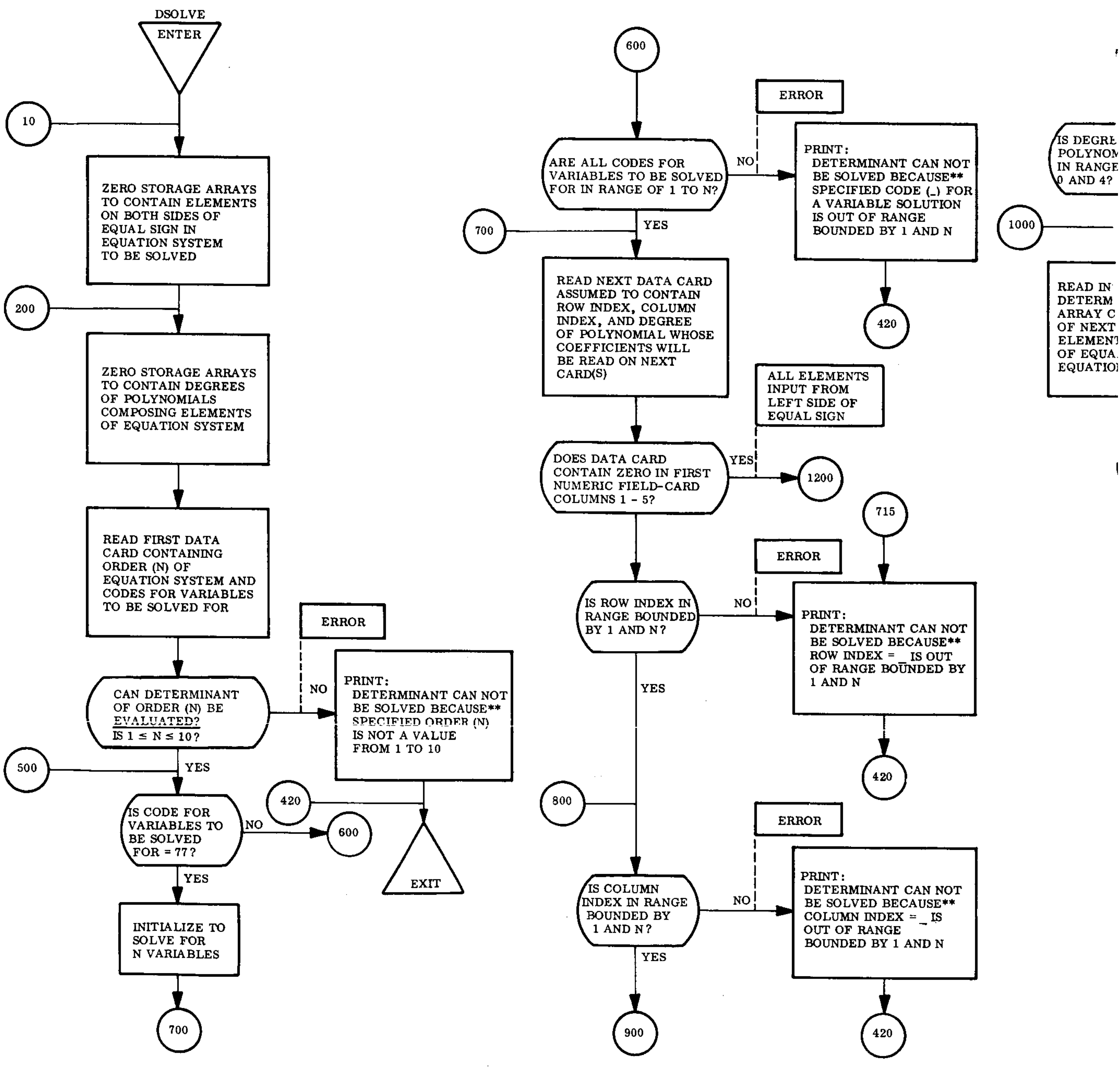


Represents an entry point to the main program or a subroutine



Represents an exit point from the main program for a subroutine

Represents a comment.






FOLDOUT FRAME

2

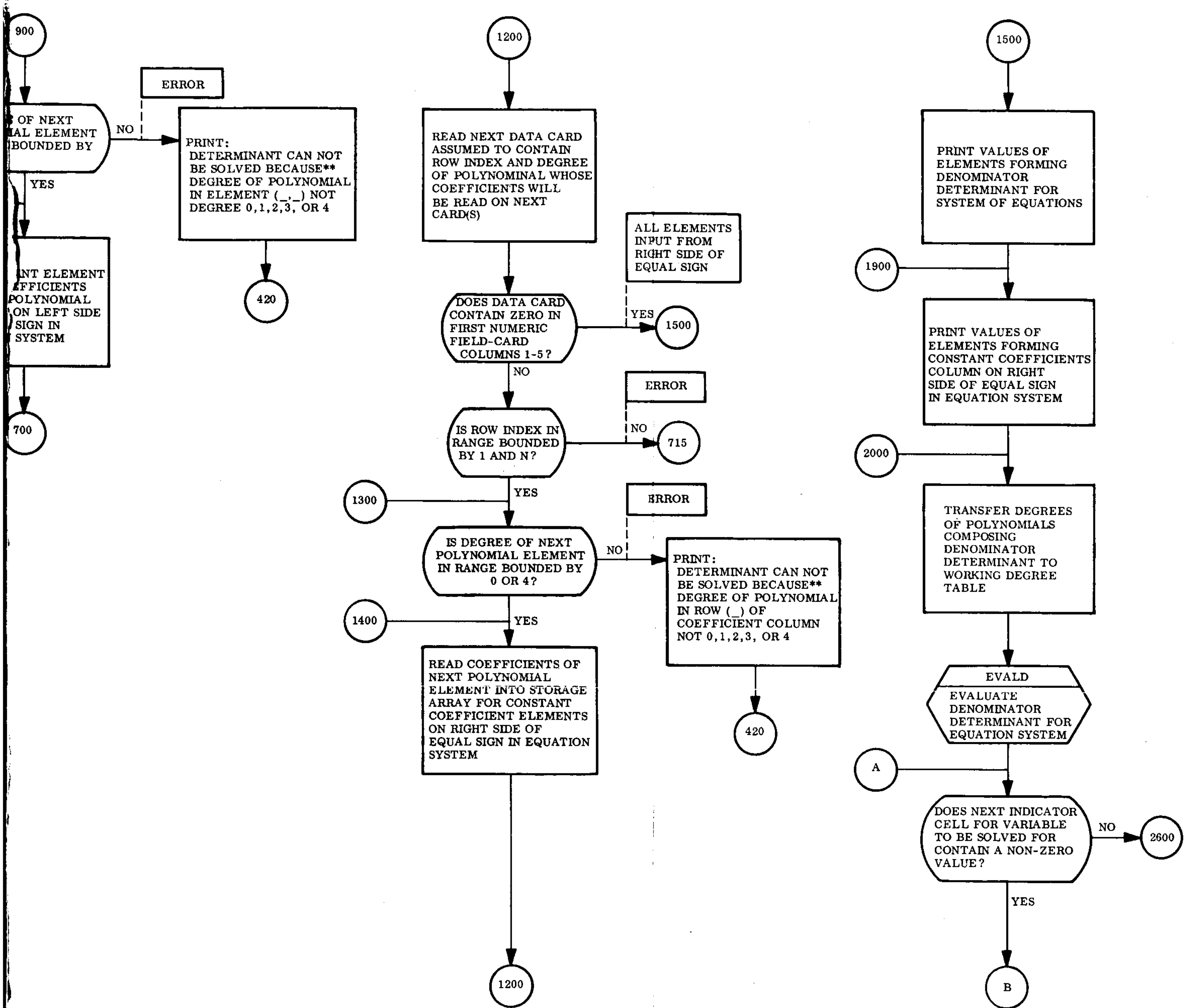
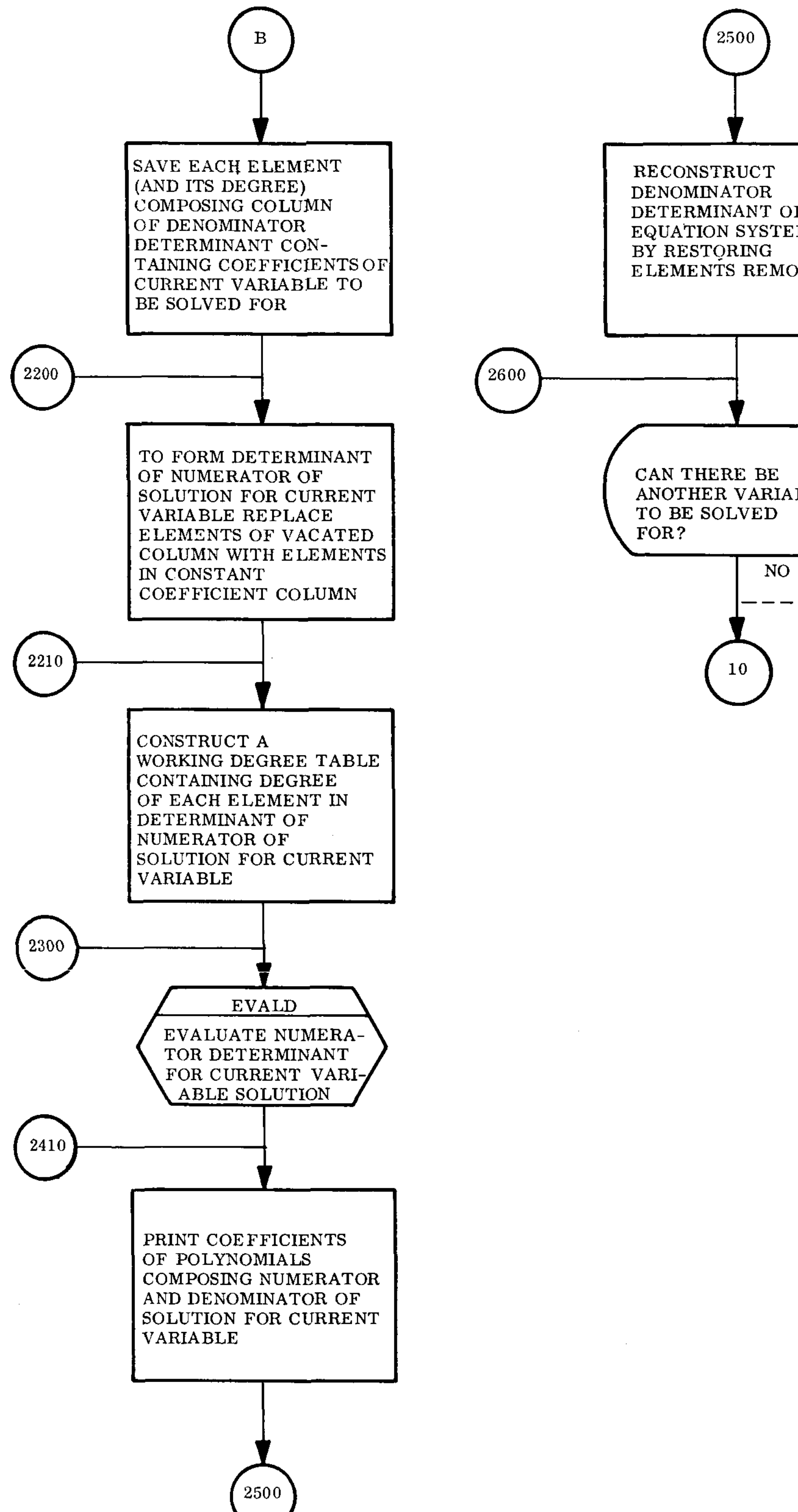



Figure B-3. Flow Chart 1 For Program DSOLVE



FOLDOUT FRAME




B-15

FOLDOUT FRAME

21

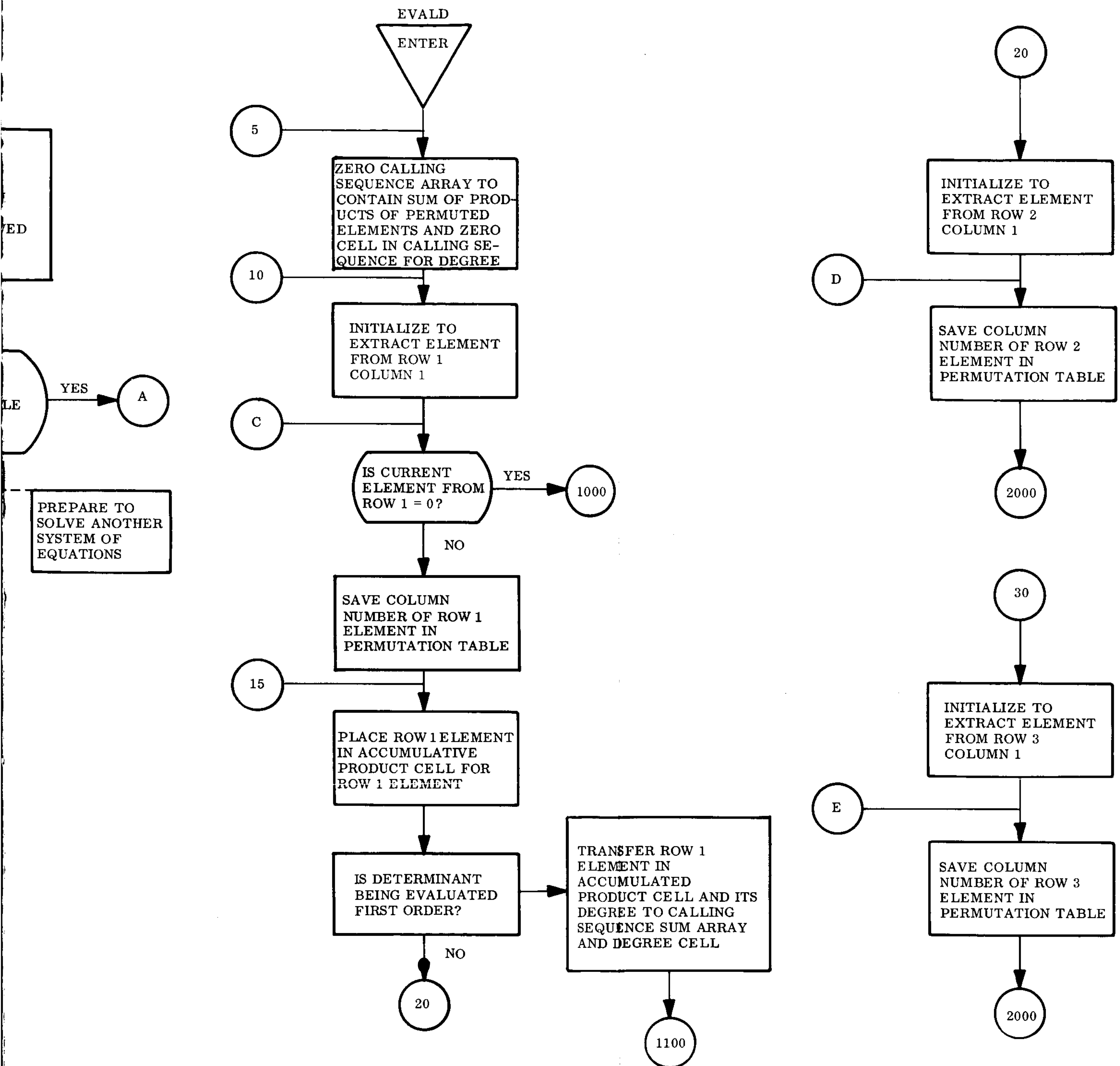
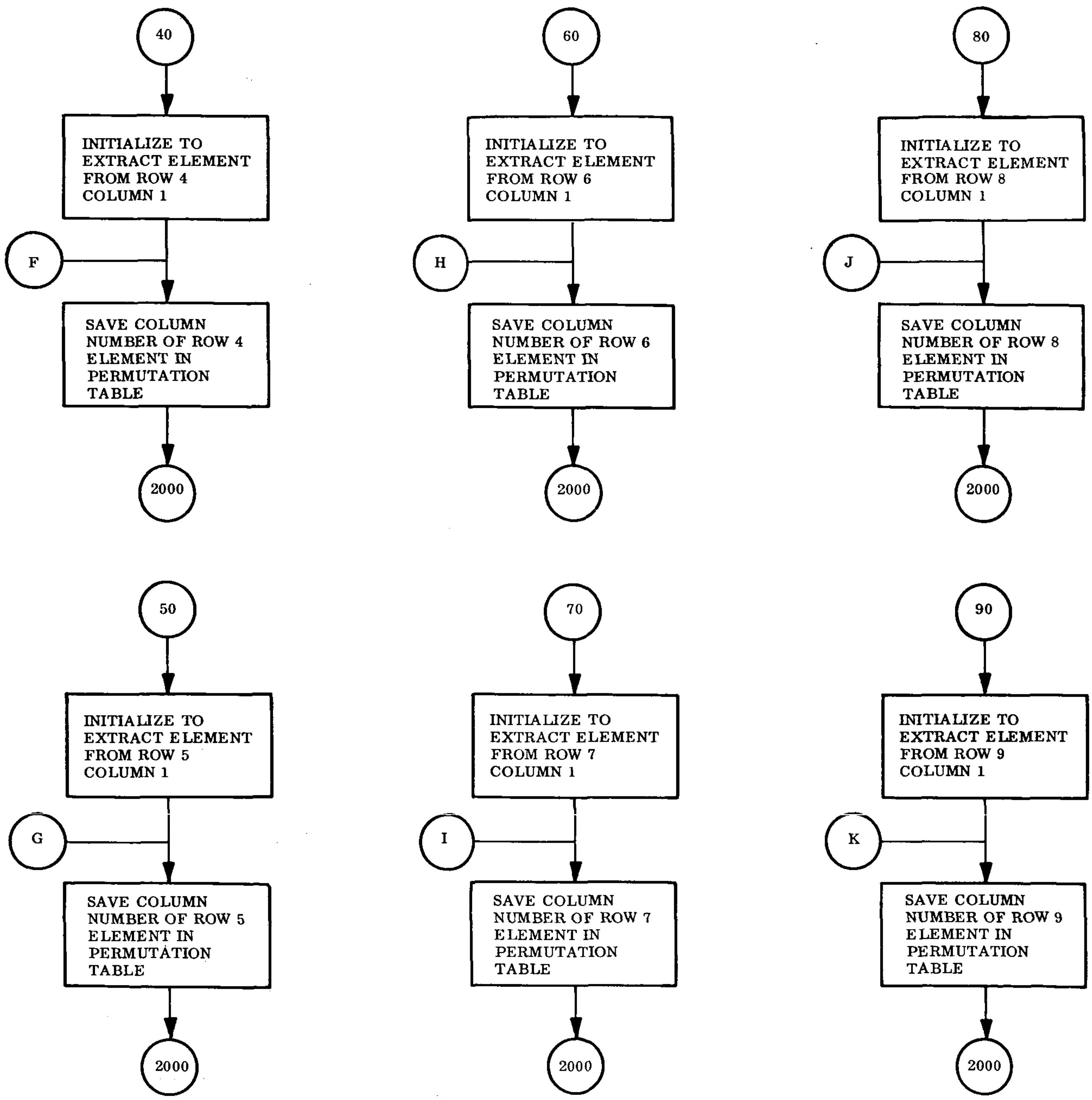



Figure B-4. Flow Chart 2 For Program DSOLVE

FOLDOUT FRAME






FOLDOUT FRAME

2

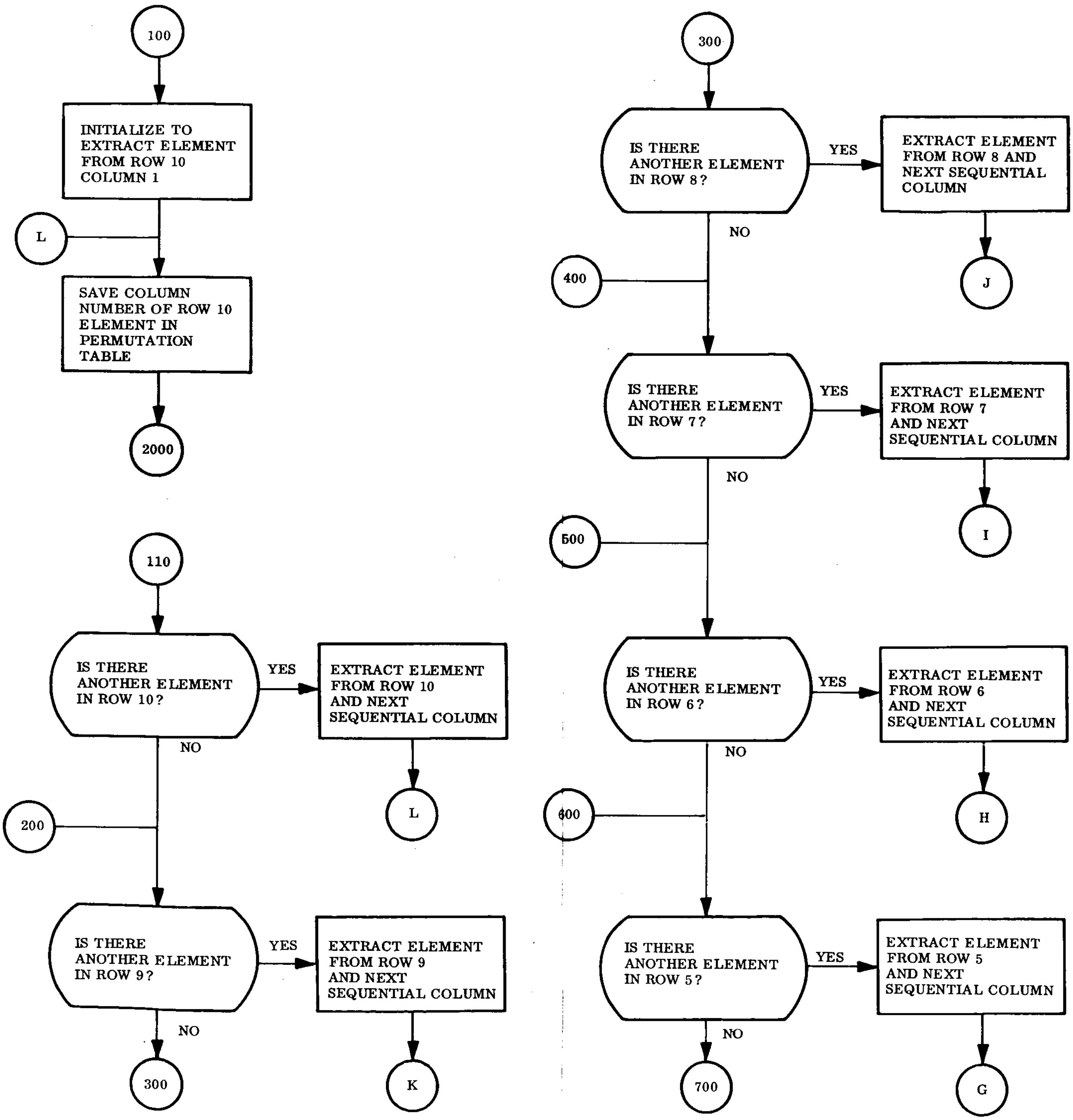
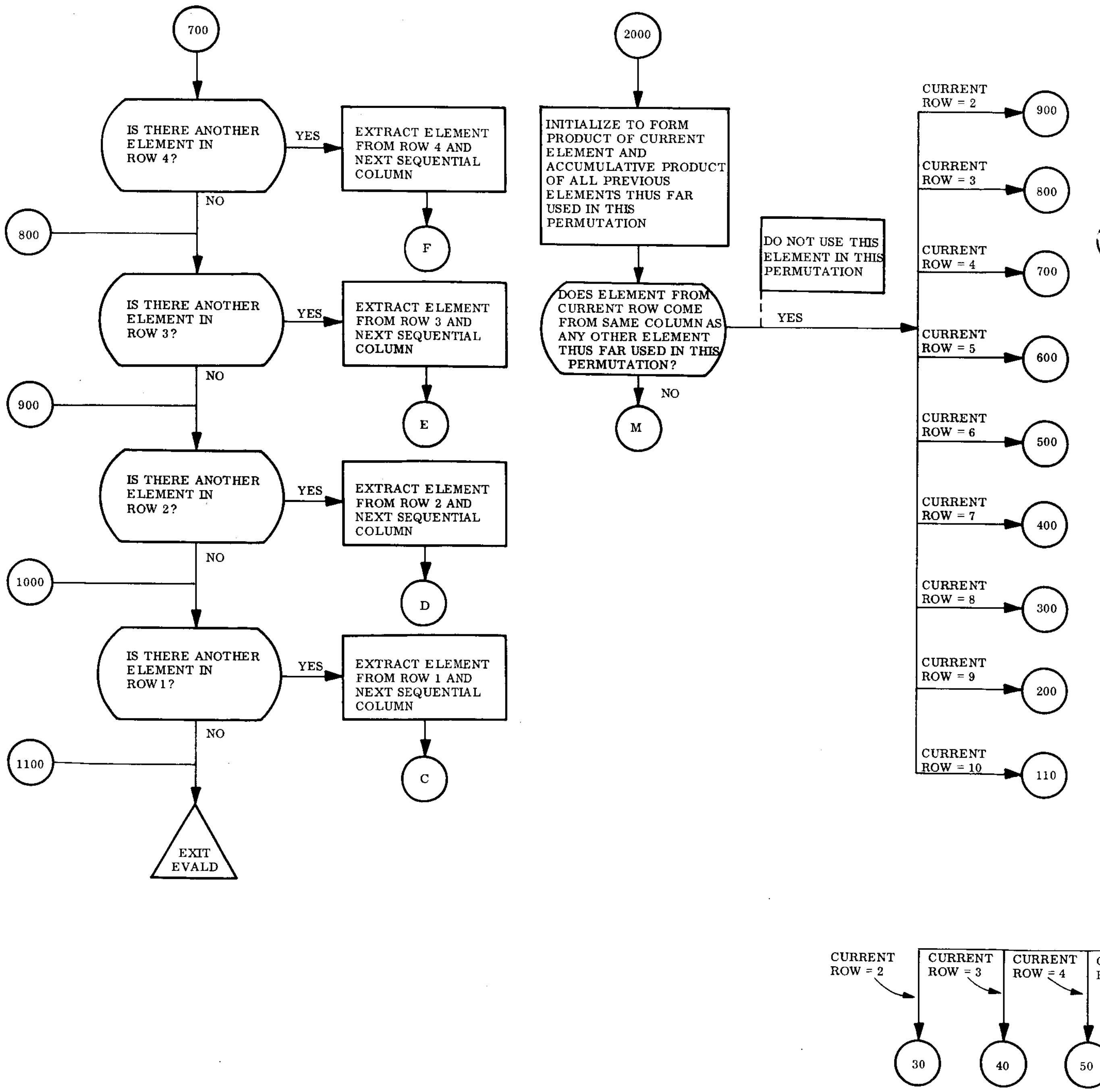



Figure B-5. Flow Chart 3 For Program DSOLVE






FOLDOUT FRAME

2

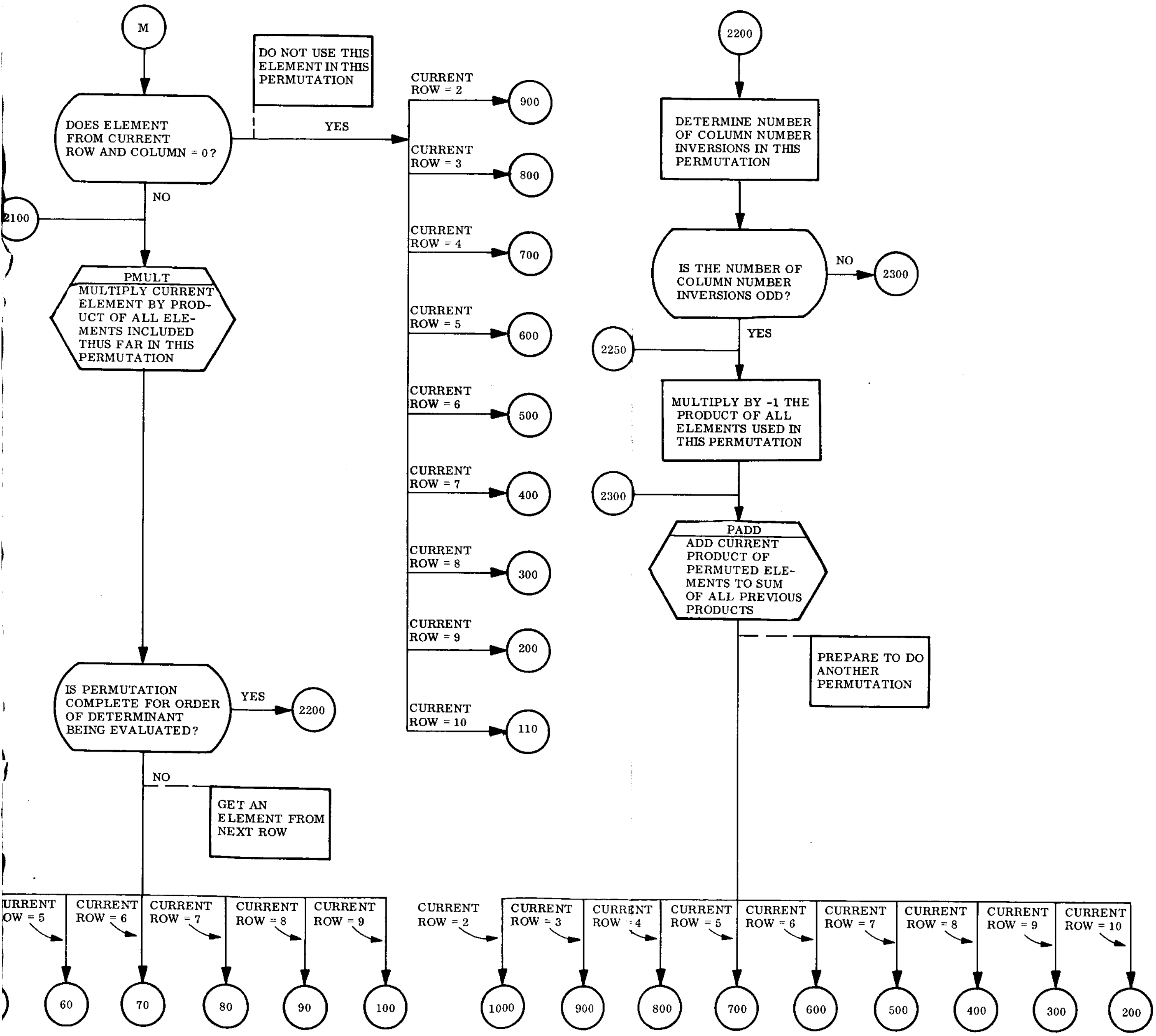
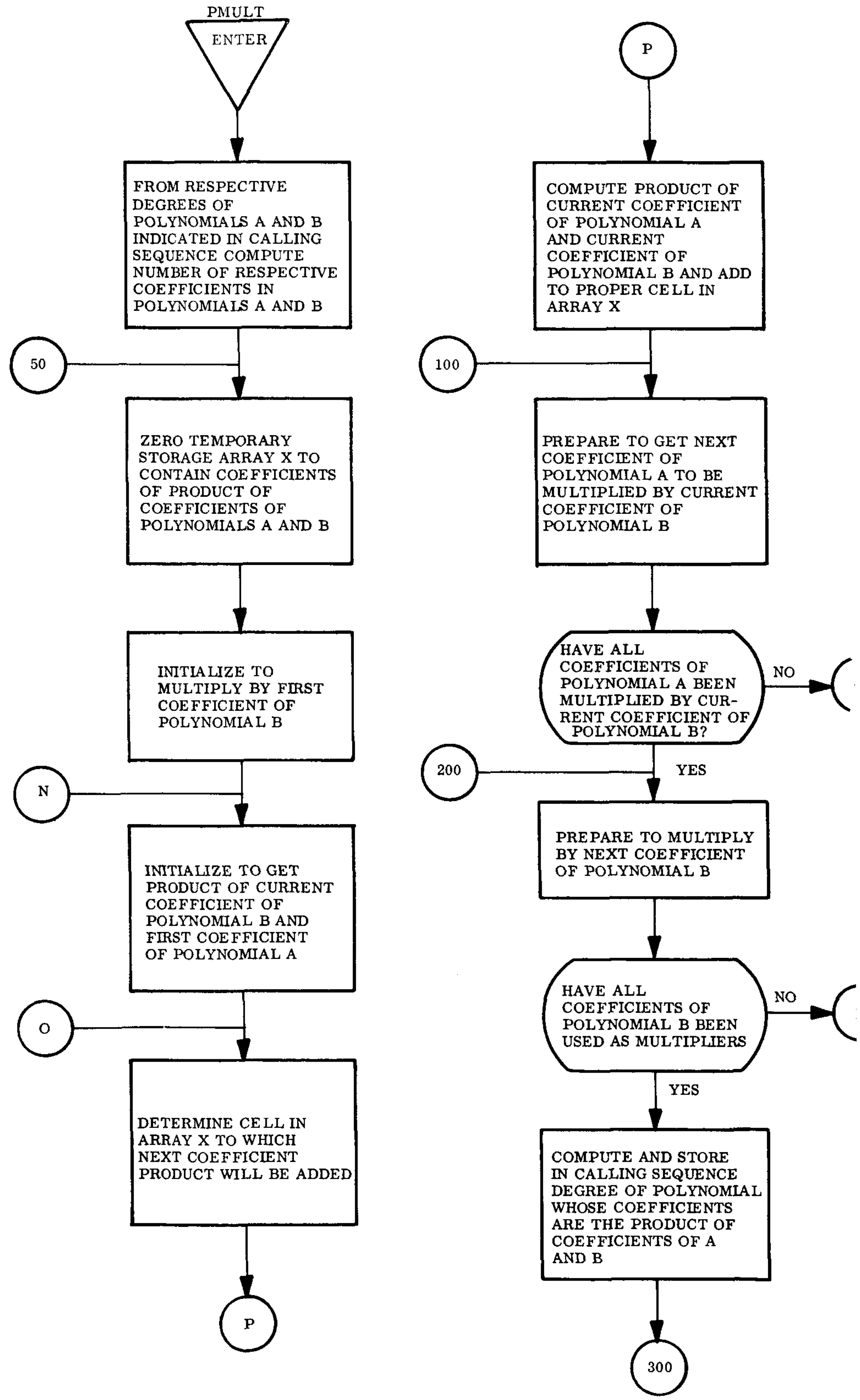



Figure B-6. Flow Chart 4 For Program DSOLVE

FOLDOUT FRAME






FOLDOUT FRAME

2

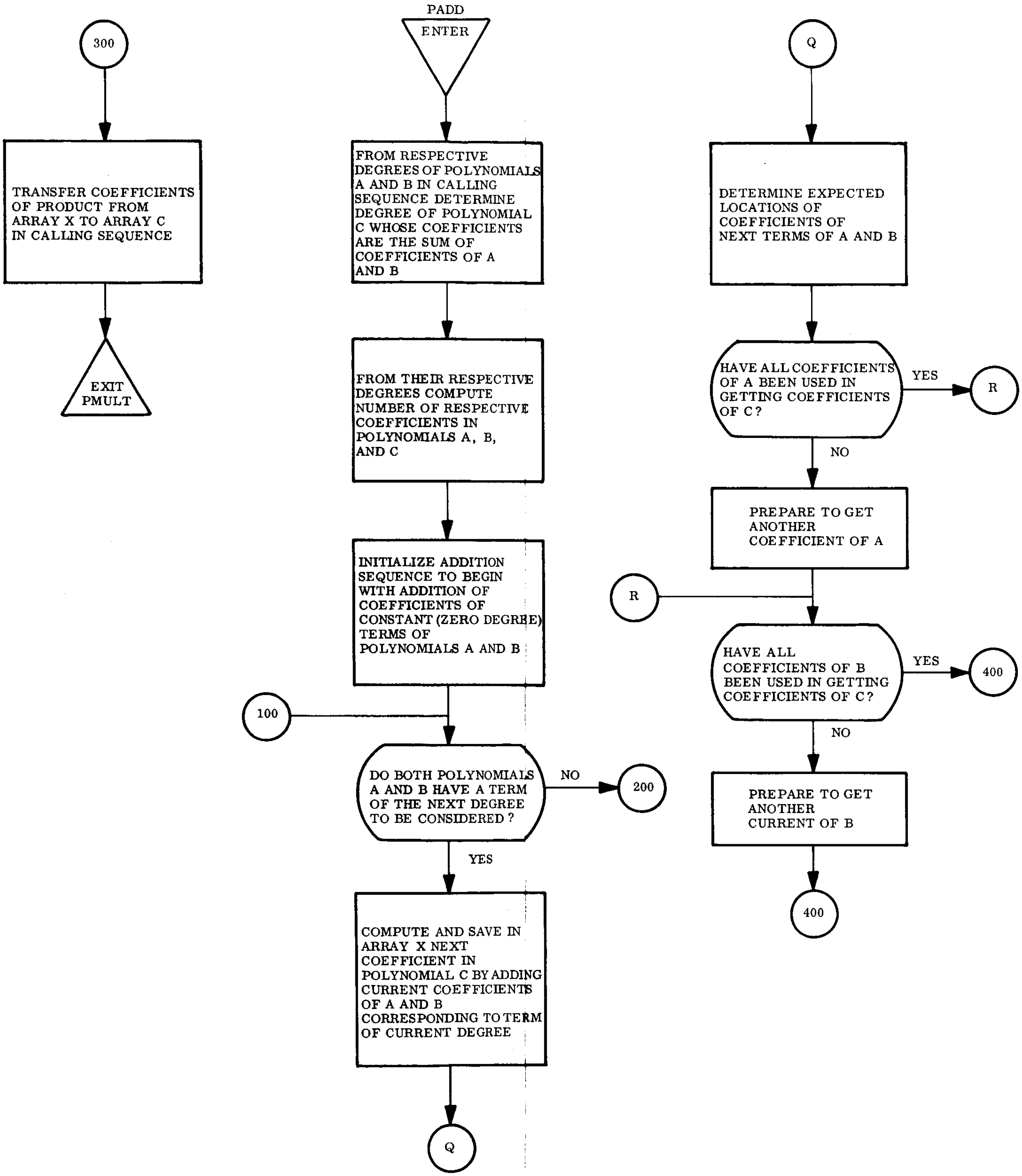



Figure B-7. Flow Chart 5 For Program DSOLVE

FOLDOUT FRAME

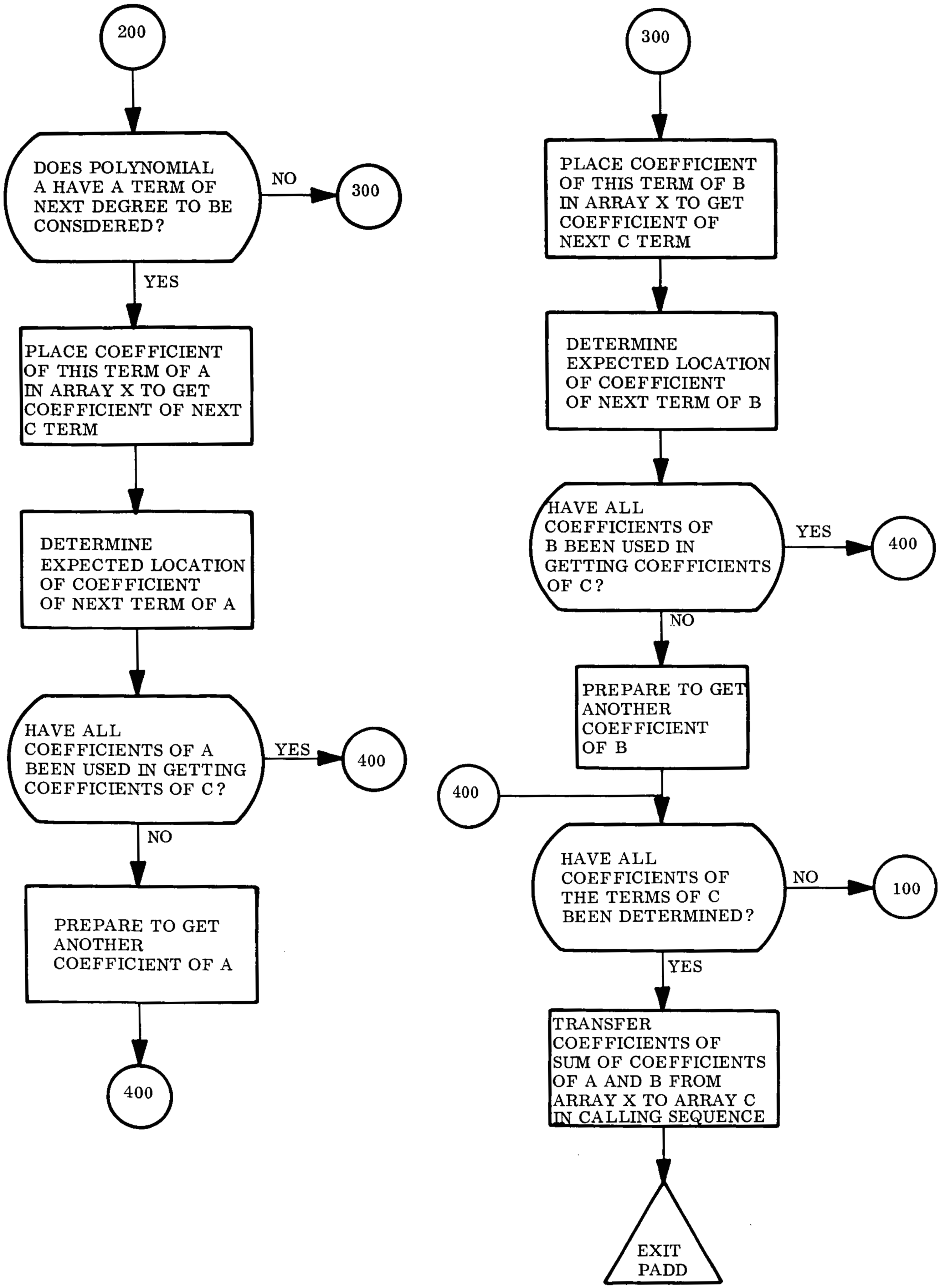



Figure B-8. Flow Chart 6 For Program DSOLVE

## APPENDIX C

## NUMERICAL DETERMINATION OF SYSTEM TRANSFER FUNCTION BY ROOT-FINDING AND CURVE-FITTING

### C.1 INTRODUCTION

Another method of determining the system transfer function is the numerical one described in Reference 1. Briefly, the method is to find the system transfer function numerator and denominator numerically for several values of the variable s, determine the real roots, and find the coefficients of the remainder polynomials by curve-fitting. Application of this method to some simple network problems yields satisfactory results.

Attempts to solve the ST-124-M problem, using the same equations as for the analytical approach were unsuccessful. The difficulties were found to be caused by the curve-fitting regression equations having coefficients differing by many orders of magnitude. Solving part of the overall problem by finding the transfer functions for $E_0/E_p$ and $E_0/E_A$ did give good results.

To overcome the difficulties encountered in the complete system solution, improvements in the method have been devised consisting of an alternate method of curve-fitting and a procedure for inverting matrices by partitioning. These are not related, and either could be used without the other. The partioning procedure has yet to be tried.

### C.2 INTERMEDIATE RESULTS

In reducing the matrices D-1 and D-2 by this technique, the following results were obtained:

$$T_{\epsilon_p,\epsilon_0} = \frac{15.44124942\,(s^4 + 343.318158s^3 + 31{,}688.4825s^2 + 461{,}059.791s + 2.54857286)}{s^5 + 555.575921s^4 + 346{,}545.402s^3 + 44{,}629{,}602.8s^2 + 696{,}723{,}329s + 983{,}056{,}705}$$

$$T_{\epsilon_A,\epsilon_0} = \frac{1.00008(s^5 + 262.569806s^4 + 17{,}575.529s^3 + 707{,}566.285s^2 + 13{,}624{,}924s + 92{,}545{,}804.38)}{s^5 + 555.575921s^4 + 346{,}545.402s^3 + 44{,}629{,}602.8s^2 + 696{,}723{,}329s + 983{,}056{,}705}$$

These compare favorably with the analytical results:

$$T_{\epsilon_p, \epsilon_0} = \frac{15.4411775s\,(s^3 + 343.3177s^2 + 31,688.468s + 461,059.107)}{s^5 + 555.660s^4 + 346,597.12s^3 + 44,636,166s^2 + 696,824,882s + 983,172,360}$$

$$T_{\epsilon_A, \epsilon_0} = \frac{s^5 + 262.438s^4 + 17,486.06s^3 + 706,702.12s^2 + 13,618,329s + 92,550,835}{s^5 + 555.660s^4 + 346,597.12s^3 + 44,636,166s^2 + 696,824,882s + 983,172,360}$$

The agreement is within the accuracies usually required for engineering work and can be made better with the adjustment of some of the numerical criteria that are used in the program.

## C.3 TRANSFER FUNCTIONS BY CURVE-FITTING

The problem to be solved can be stated as follows:

Given a linear system

$$T(s)\,\vec{Y} = \vec{V}(s) \tag{C-1}$$

where the elements of the matrix T(s) and of the vector $\vec{V}(s)$ are rational polynomials, analytic solutions for the elements $y_i$ of the vector $\vec{Y}$ are sought in the form

$$y_i = \frac{P_i(s)}{Q(s)} \quad , \qquad Q(s) = |T(s)| \quad , \tag{C-2}$$

where $P_i(s)$ and Q(s) are rational polynomials. In a linear network system, the matrix T(s) is the transfer matrix and the vector $\vec{Y}$ is the vector of the output nodes.

The solution to the above problem is to be effected by numerical matrix inversion and curve-fitting techniques. A computer program has been set up to determine the numerical value of each element of $\vec{Y}$ for assigned values of the independent variable s. Let these numerical values of the output node be denoted by $\hat{y}_i(s)$. The variation of $\hat{y}_i(s)$ from its true value, denoted by $y_i(s)$, will be a function, among other things, of the programs and the subroutines used. For instance, a double precision program or subroutine will produce less deviation than will a single precision program. At any rate, the problem is, that subject to the limitations of an existing program, what is the best possible fit of a rational function to a set of values of $\hat{y}_i(s)$, if such a rational function is specified in the form of a ratio of two rational polynomials with prescribed degrees.

There are two possible approaches to curve-fitting which can be termed linear and nonlinear.

The linear approach was described in Reference 1 and has been tested in some simple cases in which the degrees of the polynomials involved were eight and nine. The results were encouraging and the accuracy very satisfactory, in spite of the fact that the program used a single precision inversion subroutine. This program was set up to perform the following tasks:

a. Determine the various degrees of Q and each $P_i$ by analytic considerations.

b. Extract as many real roots as possible for the denominator Q(s).

c. Curve-fit the polynomial $Q_2(s)$, where

$$Q(s) = Q_1(s)Q_2(s)$$

$$Q_1(s) = (s - s_1)\ldots(s - s_r)$$

and $s_1$, $s_2$, ... $s_r$ are the roots extracted in step b.

d. Curve-fit each $P_i(s)$ using steps equivalent to b and c.

In simple cases where the polynomials involved are of moderate degrees, the linear curve-fitting approach may prove efficient and satisfactory for all purposes. In large order systems, however, in which the degrees of the various polynomials involved are large, various numerical problems may arise.

Though at this stage of the art, the merits of one approach or another cannot be fully evaluated, some possible trouble areas may be identified. In the first place, step a above was introduced to avoid the need for fitting high degree polynomials; nevertheless, the step may prove ineffective whenever there are few real roots. Second, it may be an expensive procedure time-wise, especially when this step is repeated for all numerators of the output nodes since each root extraction required 10 to 15 iterations. Third, to accommodate the curve-fitting of high degree polynomials, some control must be exercised over the range of values of s that is used. If the range is too small, the fitting may not be accurate; if too large, big numbers are involved. For these reasons, and perhaps more, such as the normalizing characteristics and the choice that can be exercised in overcoming ill-conditioned matrices, the method of curve-fitting described below deserves exploration of its full possibilities.

Thus, if in Equation C-2, the vector of the coefficients in the numerator is represented by $\vec{A}$, and that in the denominator by $\vec{B}$, where

$$\vec{A} = (a_1, \ldots, a_n)$$

$$\vec{B} = (b_1, \ldots, b_m)$$

Then the function $y_i$ becomes a function of $\vec{A}$, $\vec{B}$, and s:

$$y_i = y_i(\vec{A}, \vec{B}, s)$$

If $\vec{A}^*$ and $\vec{B}^*$ are two vectors very close to the two vectors $\vec{A}$ and $\vec{B}$ which represent the best fit, then $y_i$ can be approximated by the linear terms in the Taylor series expansion about the set $(\vec{A}^*, \vec{B}^*)$.

Then

$$y_i(\vec{A}, \vec{B}, s) = y_i(\vec{A}^*, \vec{B}^*, s) + \sum \frac{\partial y_i}{\partial a_i} da_i \left| \begin{matrix} \vec{A} = \vec{A}^* \\ \vec{B} = \vec{B}^* \end{matrix} \right| + \sum \frac{\partial y_i}{\partial b_i} db_i \left| \begin{matrix} \vec{A} = \vec{A}^* \\ \vec{B} = \vec{B}^* \end{matrix} \right| \qquad \text{(C-3)}$$

Since $da_i = a_i - a^*_i$ and $db_i = b_i - b^*_i$, Equation C-3 can be written as

$$y_i(\vec{A}, \vec{B}, s) - y_i(\vec{A}^*, \vec{B}^*, s) = \sum \frac{\partial y_i(\vec{A}^*, \vec{B}^*, s)}{\partial a^*_i} a_i + \sum \frac{\partial y_i(\vec{A}^*, \vec{B}^*, s)}{\partial b^*_i} b_i - \left[ \sum \frac{\partial y_i(\vec{A}^*, \vec{B}^*, s)}{\partial a^*_i} a^*_i + \sum \frac{\partial y_i(\vec{A}^*, \vec{B}^*, s)}{\partial b^*_i} b^*_i \right] \qquad \text{(C-4)}$$

Thus the problem of nonlinear curve-fitting has been reduced to a linear one. The function $y_i(A^*, B^*, s)$ may be identified with the computed value of $y_i$; therefore, the left hand side of Equation C-4 represents the residual error. The bracketed term on the right hand side of C-4 may be identified with what is referred to in linear curve-fitting as tabulated or observed values and can be computed. Also, the remaining partial derivatives may be identified with the regression matrix, and these can be computed.

Now the question may be raised that, in this procedure, the vector $\vec{B}$ is calculated as many times as the number of output nodes, although $\vec{B}$ should be the same in all output nodes. However, the seemingly unnecessary repetition of the calculation of $\vec{B}$ yields important information which can be used as a criterion of best possible fit. That is, if there is an unsatisfactory variation of $\vec{B}$ from one output node to another, the curve-fitting may be repeated by substituting $\vec{A}$ for $\vec{A}^*$ and $\vec{B}$ for $\vec{B}^*$ in Equation C-4 and the equations solved again. Time-wise, a repeated curve-fitting takes less time than the initial one since there are no inversions of the matrix T(s) involved.

## C.4 CURVE-FITTING BY RATIONAL FUNCTIONS

Assume that the fitting function f is of the form

$$f = \frac{a_1 s^{n-1} + a_2 s^{n-2} + \ldots + a_n}{1 + b_1 s^{n-1} + b_2 s^{n-2} + \ldots + b_{n-1} s} \tag{C-5}$$

For convenience, the same degrees for numerator and denominator have been assumed, since for the case in which the degrees of the numerator and denomiantor are m and n (n > m), a function g defined as

$$g = 1 + f \tag{C-6}$$

can always be fitted instead.

Let $\hat{f}$ denote the computed value of f:

$$\hat{f} = \frac{a_1{*}s^{n-1} + a_2{*}s^{n-2} + \ldots + a_n{*}}{1 + b_1{*}s^{n-1} + b_2{*}s^{n-2} + \ldots + b{*}_{n-1} s} \tag{C-7}$$

In terms of the coefficients of f and $\hat{f}$, two vectors $\vec{A}$ and $\vec{A}*$ are defined, thus:

$$\vec{A} = (a_1, a_2, \ldots, a_n, b, b_2, \ldots, b_{n-1})$$

$$\vec{A}* = (a_1*, a_2*, \ldots, a_n*, b_1*, b_2*, \ldots, b*_{n-1})$$

Assuming the two points represented by the two above vectors are close to each other, f can be expanded in a Taylor series about $\vec{A}*$ and only the linear terms preserved. Thus

$$f(\vec{A}) = f(\vec{A}*) + \sum_{i=1}^{n} \frac{\partial f(\vec{A})}{\partial a_i} da_i \Big| \vec{A} = \vec{A}* \tag{C-8}$$

$$+ \sum_{i=1}^{n-1} \frac{\partial f(\vec{A})}{\partial b_i} db_i \Big| \vec{A} = \vec{A}*$$

where $da_i = a_i - a^*_i$, $db_i = b_i - b^*_i$

or

$$f - \hat{f} = \frac{\sum_{i=1}^{n} s^{n-i} a_i}{d(s)} - \frac{\hat{f} \sum_{i=1}^{n-1} s^{n-i} b_i}{d(s)} - \frac{\hat{f}}{d(s)} \tag{C-9}$$

The term $f - \hat{f}$ in Equation C-9 represents the residual error, and the function d(s) is defined as:

$$d(s) = 1 + b_1^* s^{n-1} + \ldots + b^*_{n-1} s \tag{C-10}$$

It can be seen that if the following system is to be solved,

$$T(s)\vec{Y} = \vec{V}(s) \tag{C-11}$$

where T(s) is a matrix whose elements are rational polynomials, $\vec{Y}$ is the vector of the output nodes and $\vec{V}(s)$ is the constant vector with rational polynomials, then d(s) in Equation C-9 is the determinant of T(s) divided by the constant term T(0) where $T(0) \neq 0$.

If numerical values $s_1$ to $s_k$ are assigned to s, the regression coefficients of (C-9) are given by the equation.

$$X^T X \vec{A} = X^T \vec{B} \tag{C-12}$$

where

$$\vec{A} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \\ b_1 \\ b_2 \\ \vdots \\ b_{n-1} \end{bmatrix} \qquad \vec{B} = \begin{bmatrix} \frac{\hat{f}_1}{d_1} \\ \frac{\hat{f}_2}{d_2} \\ \cdot \\ \cdot \\ \cdot \\ \frac{\hat{f}_k}{d_k} \end{bmatrix} \tag{C-13}$$

$$X = \begin{bmatrix} \frac{s_1^{n-1}}{d_1}, \frac{s_1^{n-2}}{d_1} \cdots \frac{1}{d_1}, \frac{-\hat{f}_1 s_1^{n-1}}{d_1}, \ldots, \frac{-\hat{f}_1 s_1}{d_1} \\ \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\ \frac{s_k^{n-1}}{d_k}, \frac{s_k^{n-2}}{d_k} \cdots \frac{1}{d_k}, \frac{-\hat{f}_k s_1^{n-1}}{d_k}, \ldots, \frac{-\hat{f}_k s_k}{d_k} \end{bmatrix} \tag{C-14}$$

where $\hat{f}_1, \ldots, \hat{f}_k$, and $d_1, \ldots, d_k$ are the corresponding values of f and d(s) for $s_1, s_2, \ldots, s_k$ $(k > 2n - 1)$.

Since the matrix $X^T X$ can be of a large order if n is very large, partitioning techniques will be employed for the purpose of controlling computational errors more effectively.

For this purpose the matrices $X_1$, $X_2$, F, and D are defined as follows:

$$X_1 = \begin{bmatrix} s_1^{n-1} \ldots s_1, 1 \\ \vdots \qquad \vdots \\ s_k^{n-1} \ldots s_k, 1 \end{bmatrix}, \quad X_2 = \begin{bmatrix} s_1^{n-1}, \ldots s_1 \\ \vdots \qquad \vdots \\ s_k^{n-1} \quad \ldots s_k \end{bmatrix}$$

$$F = \begin{bmatrix} \hat{f}_1 & & & \\ & \hat{f}_2 & & \\ & & \ddots & \\ & & & \hat{f}_k \end{bmatrix}, \quad D = \begin{bmatrix} \frac{1}{d_1} & & & \\ & \frac{1}{d_2} & & \\ & & \ddots & \\ & & & \frac{1}{d_k} \end{bmatrix} \tag{C-15}$$

where $X_1$ is a kxn matrix,

$X_2$ is kx(n - 1)

F and D are kxk (diagonal)

Then

$$X = [DX_1, \; - DFX_2] \quad , \quad X^T = \begin{bmatrix} X_1{}^T D \\ -X_2{}^T FD \end{bmatrix} \tag{C-16}$$

so that

$$X^T X = \begin{bmatrix} X_1{}^T D^2 F X_1, & - X_1{}^T D^2 F X_2 \\ -X_2{}^T D^2 F X_1, & X_2{}^T D^2 F^2 X_2 \end{bmatrix}$$

Denote the above partitioning as follows

$$X^T X = \begin{array}{|c|c|} \hline Y_1 & Y_2 \\ \hline Y_3 & Y_4 \\ \hline \end{array} \tag{C-17}$$

in which the orders are

| | |
|---|---|
| n.n | n.(n - 1) |
| (n - 1).n | (n - 1).(n - 1) |

Both square matrices $Y_1$ and $Y_4$ are symmetrical matrices and

$$Y_3 = Y_2{}^T$$

The matrices $Y_1$, $Y_2$, $Y_3$, and $Y_4$ in their explicit forms are:

$$Y_1 = \begin{bmatrix} \sum \frac{s_i^{2n-2}}{d_i^2}, & \ldots, & \sum \frac{s_i^{n-1}}{d_i^2} \\ \vdots & & \vdots \\ \sum \frac{s_i^{n-1}}{d_i^2}, & \ldots, & \sum \frac{1}{d_i^2} \end{bmatrix} \tag{C-18}$$

$$Y_2 = - \begin{bmatrix} \sum \frac{\hat{f}_i s_i^{2n-2}}{d_i^2}, & \ldots, & \sum \frac{\hat{f}_i s_i^{n}}{d_i^2} \\ \vdots & & \vdots \\ \sum \frac{\hat{f}_i s_i^{n-1}}{d_i^2}, & \ldots, & \sum \frac{\hat{f}_i s_i}{d_i^2} \end{bmatrix} \tag{C-19}$$

$$Y_3 = Y_2^T \tag{C-20}$$

$$Y_4 = \begin{bmatrix} \sum \frac{\hat{f}_i^2 s_i^{2n-2}}{d_i^2}, & \ldots, & \sum \frac{\hat{f}_i^2 s_i^{n}}{d_i^2} \\ \vdots & & \vdots \\ \sum \frac{\hat{f}_i^2 s_i^{n}}{d_i^2}, & \ldots, & \sum \frac{\hat{f}_i^2 s_i^{2}}{d_i^2} \end{bmatrix} \tag{C-21}$$

Comparing corresponding elements:

$$Y_1: \text{ (ij) Element } = \sum_{r=1}^{k} \frac{s_r^{2n - (i+j)}}{d_r^2}$$

$$Y_2: \text{ (ij) Element } = \sum_{r=1}^{k} \frac{\hat{f}_r}{d_r^2} s_r^{2n - (i+j)} \qquad \text{(C-22)}$$

$$Y_3 = Y_2^T$$

$$Y_4: \text{ (ij) Element } = \sum_{r=1}^{k} \frac{\hat{f}_r^2}{d_r^2} s_r^{2n - (i+j)}$$

Obviously, in calculating any of the above matrices, it is not necessary to calculate all the elements because of the symmetric properties. For, if the dimension of any of the above matrices in Equation C-22 are (n x m) there are only n + m - 1 distinct elements. To build any of the matrices in Equation C-22, it suffices to calculate three vectors $V_1$, $V_2$, and $V_4$ for the matrices $Y_1$, $Y_2$, and $Y_4$.

Then, if $V_i$ (m) denotes the m'th element of the vector $V_i$ (i = 1, 2, 4),

$$V_1(m) = \sum_{r=1}^{k} \frac{s_r^{2n-1-m}}{d_r^2} \qquad m = 1, \ldots 2n-1$$

$$V_2(m) = -\sum_{r=1}^{k} \frac{\hat{f}_r}{d_r^2} s_r^{2n-1-m} \qquad m = 1, \ldots 2n-1 \qquad \text{(C-23)}$$

$$V_4(m) \sum_{r=1}^{k} \frac{\hat{f}_r^2}{d_r^2} s_r^{2n-1-m} \qquad m = 1, \ldots 2n-2$$

Then the (ij) element of $Y_i$ (i = 1, 2, 4) occupies the i + j - 1 position in its corresponding vector $V_i$.

Also, the calculation $X^T\vec{B}$ on the right hand side of Equation C-12 has been done implicitly in the vectors $V_2$ and $V_4$.

Thus

$$\text{i'th element of } X^T\vec{B} = -V_2(n+i-1),\ 1 \le i \le n \qquad \text{(C-24)}$$

$$= -V_4(i-1),\ n < i \le 2n-1$$

It should be remarked that the numbers $d_r^2$ act as normalizing factors. Also caution should be exercised in selecting the points $s_i$ so that no point lies within a small neighborhood of a zero of d(s), a selection which can be controlled.

The formal solution of Equation C-12 is

$$\vec{A} = (X^T X)^{-1} X^T \vec{B} \qquad \text{(C-25)}$$

To accomplish the inversion of $X^T X$ using partitioning, let

$$Z = (X^T X)^{-1} = \begin{array}{|c|c|} \hline Z_1 & Z_2 \\ \hline Z_3 & Z_4 \\ \hline \end{array} \qquad \text{(C-26)}$$

in which the orders are:

| | |
|---|---|
| n.n | n.(n - 1) |
| (n - 1).n | (n - 1).(n - 1) |

Multiplying Equations C-17 and C-26, and denoting by $I_j$ the jxj unit matrix and by $O_{ij}$ the ixj null matrix, the following relations are obtained:

$$\begin{aligned} Y_1 Z_1 + Y_2 Z_3 &= I_n \\ Y_1 Z_2 + Y_2 Z_4 &= O_{n.(n-1)} \\ Y_3 Z_1 + Y_4 Z_3 &= O_{(n-1).n} \\ Y_3 Z_2 + Y_4 Z_4 &= I_{n-1} \end{aligned} \qquad \text{(C-27)}$$

If $Y_1$ is nonsingular and not ill-conditioned, the solution to system C-27 is

$$\begin{aligned} Z_1 &= Y_1^{-1} - Y_1^{-1} Y_2 Z_2^T \\ Z_2 &= - Y_1^{-1} Y_2 Z_4 \\ Z_3 &= Z_2^T \\ Z_4 &= [Y_4 - Y_3 Y_1^{-1} Y_2]^{-1} \end{aligned} \qquad \text{(C-28)}$$

An alternative solution, based on the nonsingularity of $Y_4$, is

$$\begin{aligned} Z_1 &= [Y_1 - Y_2 Y_4^{-1} Y_3]^{-1} \\ Z_2 &= - Z_1 Y_2 Y_4^{-1} \\ Z_3 &= Z_2^T \\ Z_4 &= Y_4^{-1} - Y_4^{-1} Y_3 Z_2 \end{aligned} \qquad \text{(C-29)}$$

In the case that both $Y_1$ and $Y_4$ are nonsingular, both solutions C-28 and C-29 are equivalent. But if $Y_1$ and $Y_4$ are both singular, solutions C-28 and C-29 can be generalized for any arbitrary constants $\alpha$ and $\beta$ to solutions C-30 and C-31:

$$\begin{aligned} Z_1 &= (Y_1 + \alpha Y_2 Y_3)^{-1} [I_n - (Y_2 + \alpha Y_2 Y_4) Z_3] \\ Z_2 &= (Y_1 + \beta Y_2 Y_3)^{-1} [\beta Y_2 - (Y_2 + \beta Y_2 Y_4) Z_4] \\ Z_3 &= - [Y_4 - Y_3 (Y_1 + \alpha Y_2 Y_3)^{-1} (Y_2 + \alpha Y_2 Y_4)]^{-1} Y_3 (Y_1 + \alpha Y_2 Y_3)^{-1} \\ Z_4 &= [Y_4 - Y_3 (Y_1 + \beta Y_2 Y_3)^{-1} (Y_2 + \beta Y_2 Y_4)]^{-1} [I_{n-1} - \beta Y_3 (Y_1 + \beta Y_2 Y_3)^{-1} Y_2] \end{aligned} \qquad \text{(C-30)}$$

$$
\begin{aligned}
Z_1 &= [Y_1 - Y_2(Y_4 + \beta Y_3 Y_2)^{-1}(Y_3 + \beta Y_3 Y_1)]^{-1}[I_n - \beta Y_2(Y_4 + \beta Y_3 Y_2)^{-1} Y_3] \\
Z_2 &= -[Y_1 - Y_2(Y_4 + \alpha Y_3 Y_2)^{-1}(Y_3 + \alpha Y_3 Y_1)]^{-1} Y_2(Y_4 + \alpha Y_3 Y_2)^{-1} \\
Z_3 &= (Y_4 + \beta Y_3 Y_2)^{-1}[\beta Y_3 - (Y_3 + \beta Y_3 Y_1) Z_1] \\
Z_4 &= (Y_4 + \alpha Y_3 Y_2)^{-1}[I_{n-1} - (Y_3 + \alpha Y_3 Y_1) Z_2]
\end{aligned}
\tag{C-31}
$$

That these are solutions of Equations C-27 independent of $\alpha$ and $\beta$ can be shown by direct substitution. By properly choosing values of $\alpha$ and $\beta$, singular and ill-conditioned matrices can be avoided.

It should be noted that solutions C-30 and C-31 reduce to C-28 and C-29 if $\alpha = \beta = 0$.

Thus the matrix

$$
\begin{array}{|c|c|}
\hline
Z_1 & Z_2 \\
\hline
Z_3 & Z_4 \\
\hline
\end{array}
$$

is the inverse of the matrix

$$
\begin{array}{|c|c|}
\hline
Y_1 & Y_2 \\
\hline
Y_3 & Y_4 \\
\hline
\end{array}
$$

The elements $Z_1$, $Z_2$, $Z_3$, $Z_4$ are uniquely determined and are independent of the constants $\alpha$ and $\beta$. The advantage of arbitrarily choosing the constants $\alpha$ and $\beta$ without changing the Z's need hardly be further emphasized. The problem of well-conditioning the various matrices which are to be inverted is thus well under control.

Referring to Equation C-25, let $\vec{C}$ denote the vector $X^T\vec{B}$:

$$\vec{C} = X^T\vec{B} \tag{C-32}$$

and partition $\vec{C}$ into two vectors $\vec{C}_1$ and $\vec{C}_2$:

$$\vec{C} = \begin{bmatrix} \vec{C}_1 \\ \vec{C}_2 \end{bmatrix} \tag{C-33}$$

so that

$$\vec{C}_1 = \begin{bmatrix} C_1 \\ \cdot \\ \cdot \\ \cdot \\ C_n \end{bmatrix} \qquad \vec{C}_2 = \begin{bmatrix} C_{n+1} \\ \cdot \\ \cdot \\ \cdot \\ C_{2n-1} \end{bmatrix} \tag{C-34}$$

Then

$$\begin{aligned} \vec{A} &= (X^TX)^{-1}X^T\vec{B} \\ &= Z\vec{C} \\ &= \begin{bmatrix} Z_1 & Z_2 \\ Z_3 & Z_4 \end{bmatrix} \cdot \begin{bmatrix} \vec{C}_1 \\ \vec{C}_2 \end{bmatrix} \\ &= \begin{bmatrix} Z_1\vec{C}_1 + Z_2\vec{C}_2 \\ Z_3\vec{C}_1 + Z_4\vec{C}_2 \end{bmatrix} \end{aligned}$$

$$\vec{A} = \begin{bmatrix} a_1 \\ \cdot \\ \cdot \\ \cdot \\ a_n \\ b_1 \\ \cdot \\ \cdot \\ \cdot \\ b_{n-1} \end{bmatrix}$$

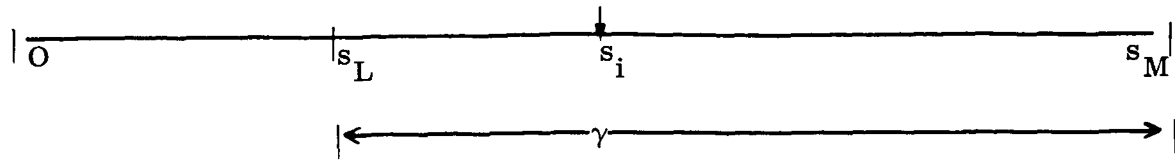
Clearly, then

$$\begin{bmatrix} a_1 \\ \\ \\ \cdot \\ \cdot \\ \cdot \\ a_n \end{bmatrix} = Z_1\vec{C}_1 + Z_2\vec{C}_2, \quad \begin{bmatrix} b_1 \\ \\ \\ \cdot \\ \cdot \\ \cdot \\ b_{n-1} \end{bmatrix} = Z_3\vec{C}_1 + Z_4\vec{C}_2 \tag{C-35}$$

The merits of applying partitioning are evident. The fact that the problem is reduced to inverting smaller order matrices plus the various choices offered in doing so are desirable features in coping with round-off errors which might show up otherwise. In addition, the operations indicated in the analysis above can be ordered in such a manner that use can be made of equivalence to a maximum degree, resulting in less storage requirements.

However, the recovery of the analytic function through numerical techniques does not depend exclusively on inverting the regression matrix as accurately as possible. Due consideration to the behavior of the fitting function in the chosen interval must be given. Thus, it may be unwise to be primarily motivated by getting a well-behaved regression matrix when choosing interval points for a fit. For instance, one cannot expect a good fit in an interval if the function to be fitted (say 10th degree polynomial) has essentially a linear behavior. On the other hand, the regression matrix may become completely ill-conditioned without some normalizing techniques, expecially when large powers are involved. In order, therefore, to overcome such difficulties, the curve-fitting procedure may be set up as follows.

Let the points $s_i$, to be used in the fitting, lie in the interval $[s_L, s_M]$




let

$$s_i = \gamma z_i \tag{C-36}$$

where

$$\gamma = s_M - s_L$$

$$z_i = \frac{s_i - s_L}{\gamma} + \delta$$

$$\delta = s_L/\gamma$$

Thus

$$z_L = \delta$$

$$z_M = 1 + \delta$$

Let the matrix U be defined as the same function of $z_i$ that X (from Equation C-12) is of $s_i$, and let the matrix W be defined as

$$W = \begin{bmatrix} \gamma^{n-1} & & & & & & & \\ & \gamma^{n-2} & & & & & & \\ & & \ddots & & & & & \\ & & & 1 & & & & \\ & & & & \gamma^{n-1} & & & \\ & & & & & \ddots & & \\ & & & & & & & \gamma \end{bmatrix} \tag{C-37}$$

The Equation C-12 becomes

$$U^T U W \vec{A} = U^T \vec{B} \tag{C-38}$$

whose solution is

$$\vec{A} = W^{-1}(U^T U)^{-1} U^T \vec{B} \tag{C-39}$$

By this procedure, for large intervals ($s_L$, $s_M$), smaller numbers will be handled in the regression matrix. In an attempt to obtain the best behaved regression matrix, it is suggested that

$$s_L = - s_M, \qquad s_M > 0$$

so that

$$\delta = - 0.5$$

and

$$-0.5 \geq z_i \geq 0.5$$

## C.5 COMPUTER FLOW CHARTS

A program for solving the matrix Equation C-1 by numerical means and curve-fitting linear functions has been prepared. Flow charts for this are shown in Figures C-1 through C-4. The implementation of these steps requires various subroutines, which are:

Subroutine Eval (N, s, T, V, R1)
Subroutine Deval (N, T, I, J, F)
Subroutine Vert (N, ADEM, C, DET)

Subroutine Eval has the following functions: When R1 = 0, it is to read the coefficients of the polynomials of the elements of T and V which are supplied as inputs and return without evaluating T or V. When R1 $\neq$ 0, call to Eval returns T and V evaluated for a specified value of s. In addition, the elements of T and V have been multiplied by R1.

Subroutines Deval and Vert do not appear in the functional chart but exist in the computer program.

The significance of the arguments of I and J is shown in the following when

| | |
|---|---|
| $I = 0, J = 0$ | $F = \|T\|$ |
| $1 \leq I, J \leq N$ | $F = (-1)^{I + J}$ cofactor of (IJ) element of T. |

Subroutine Vert serves to invert the regression matrix in the curve-fitting process. IDEM equals the dimensionality of C, the matrix being inverted, in the calling program, and DET = $|C|$. It returns with the inverted matrix of C stored in the same cells of the array C.

The present computer program is not yet final. It has been implemented and improved along the way with the purpose of overcoming the various numerical difficulties encountered. There are certain features that must be included in the program. A double precision inversion subroutine for the regression matrix should be available, but it may prove insufficient, especially in large degree polynomial curve-fitting. The monotonic behavior of the regression matrix along the diagonal plus the slanted arrangement of its elements suggest that this is an ideal situation for the application of partitioning techniques, as described. Also, double precision variables, especially in summing the various powers for building the regression matrix, are features which must be incorporated. The most important point is that sufficient evaluation of the functions to be fitted is necessary. This means that the interval of fit must be sufficiently large, yet not so large that an ill-conditioned regression matrix results. To effect a compromise in this case, the values of s in that interval must be normalized as described.

## EXPLANATION OF FLOW CHARTS

| | | |
|---|---|---|
| E1 | = | Criterion exit for degree calculation |
| E2 | = | Rounding off the degree to an integer |
| E3 | = | Criterion exit for roots |
| E4 | = | 0 or 1 in E4*EX |
| N | = | Number of output nodes |
| EX | = | 1/N |
| (SL, SM) | = | Fitting interval |
| SO | = | Initial value of S in degree finding |
| X1 | = | Initial prediction in root finding |
| NQ | = | Degree of denominator |
| NY(I) | = | Degree of numerator of output node I |
| NDEG | = | 0 for skipping degree finding procedure |
| E5 | = | Criterion for testing whether a normal exit really produced a root |
| R1 | = | 0 (1) do a new (same) problem |
| CFT | = | Normalization factor for the regression matrix |
| NRT | = | 0 for skipping root-finding procedure |

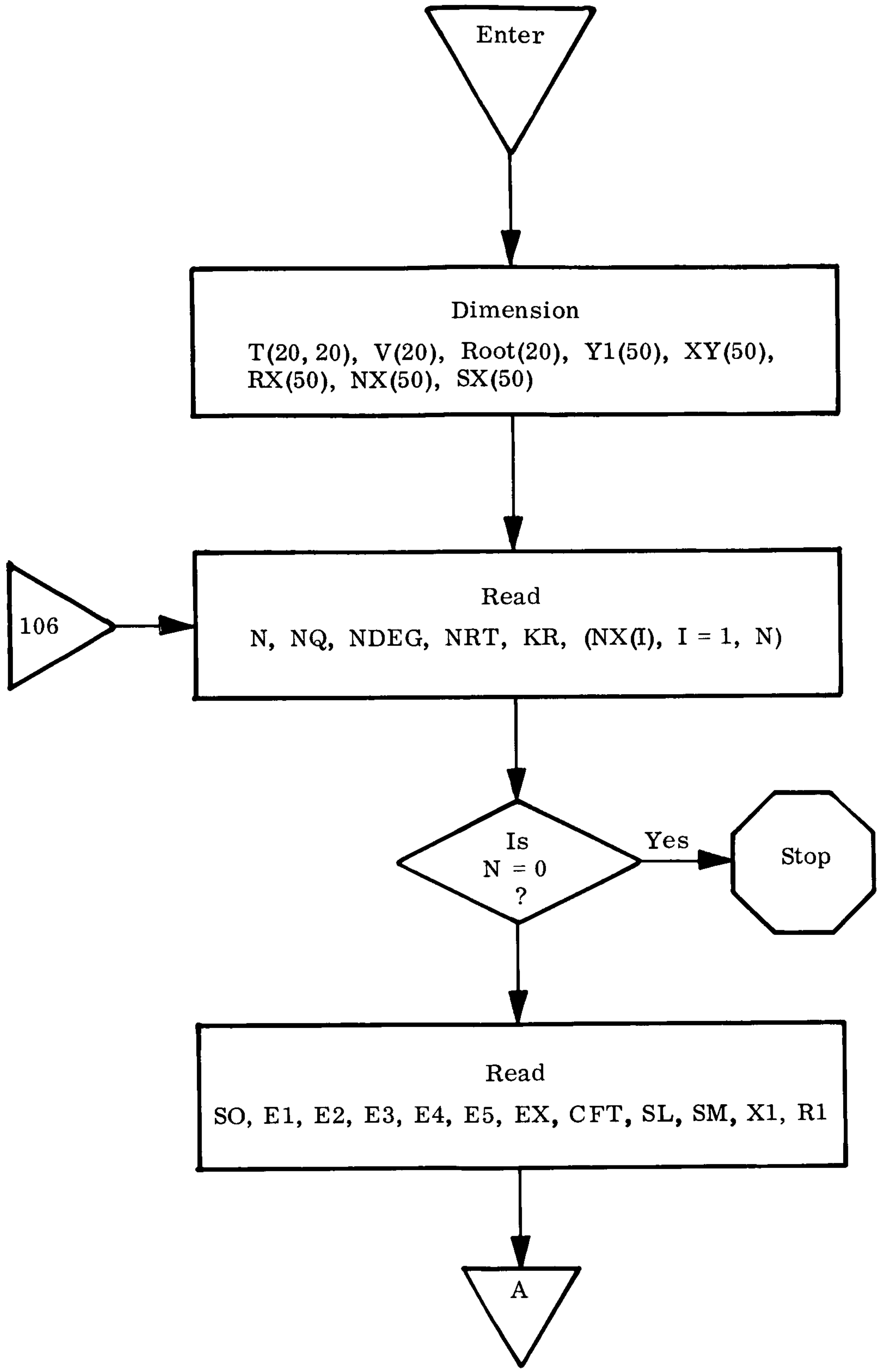



Figure C-1. Flow Chart 1 For Program CURFIT

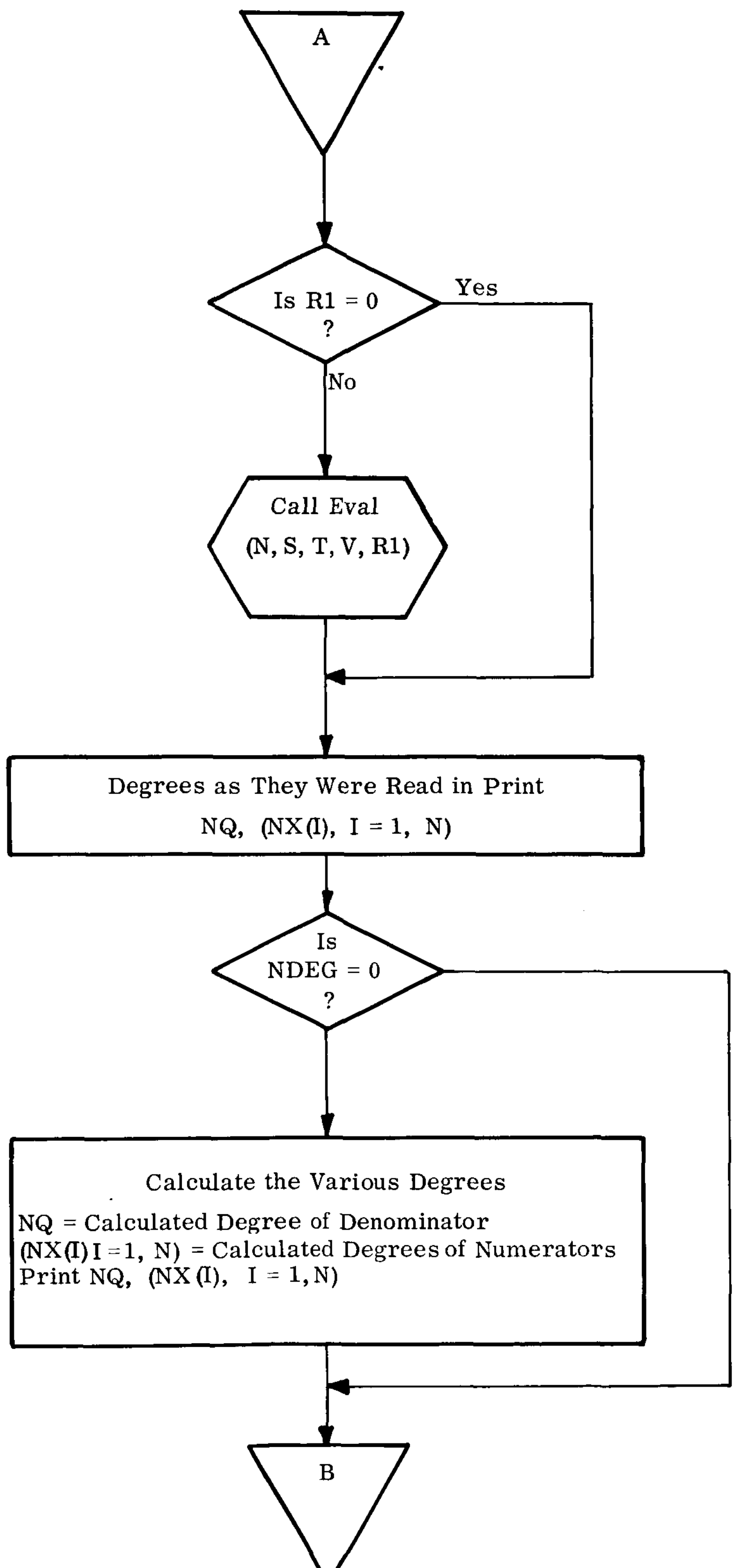



Figure C-2. Flow Chart 2 For Program CURFIT

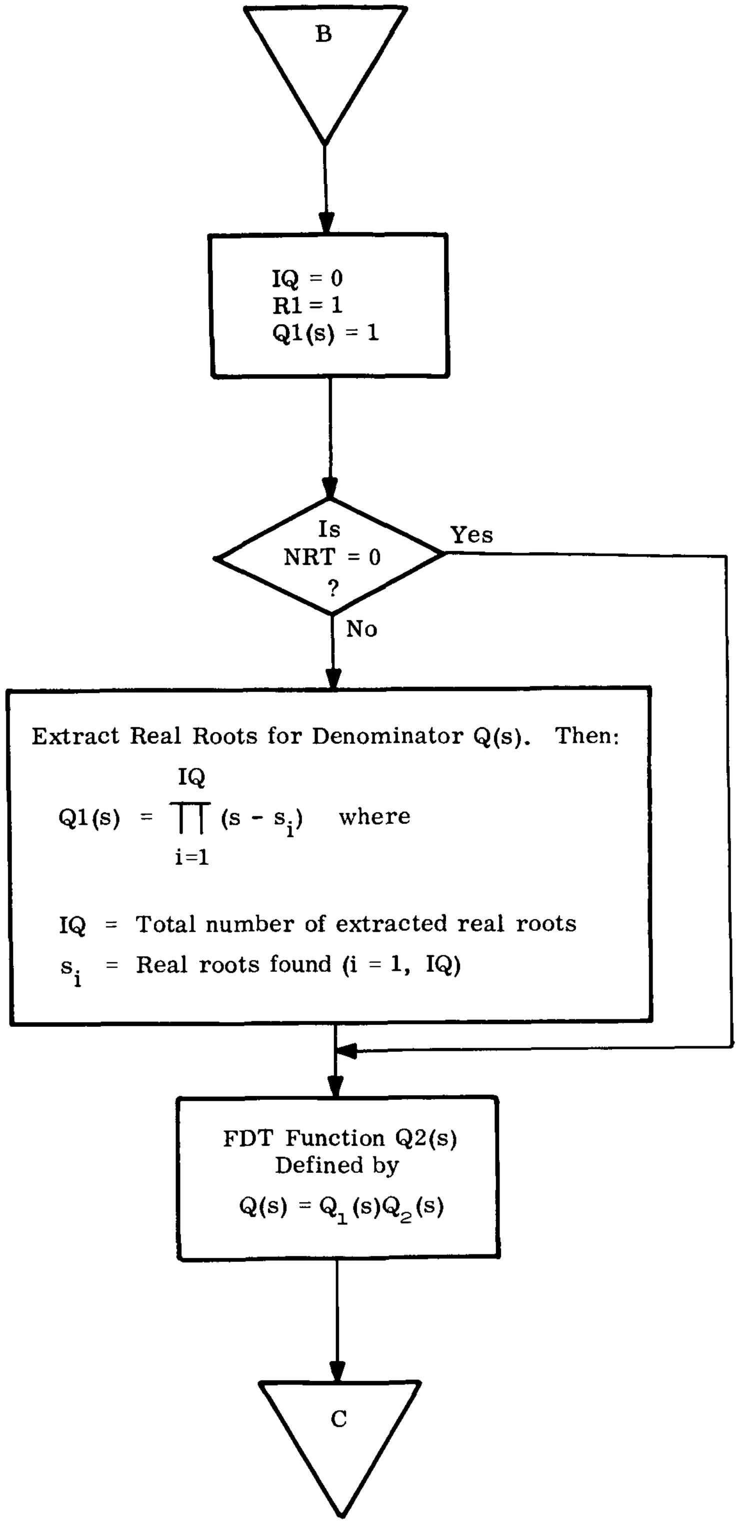



Figure C-3. Flow Chart 3 For Program CURFIT

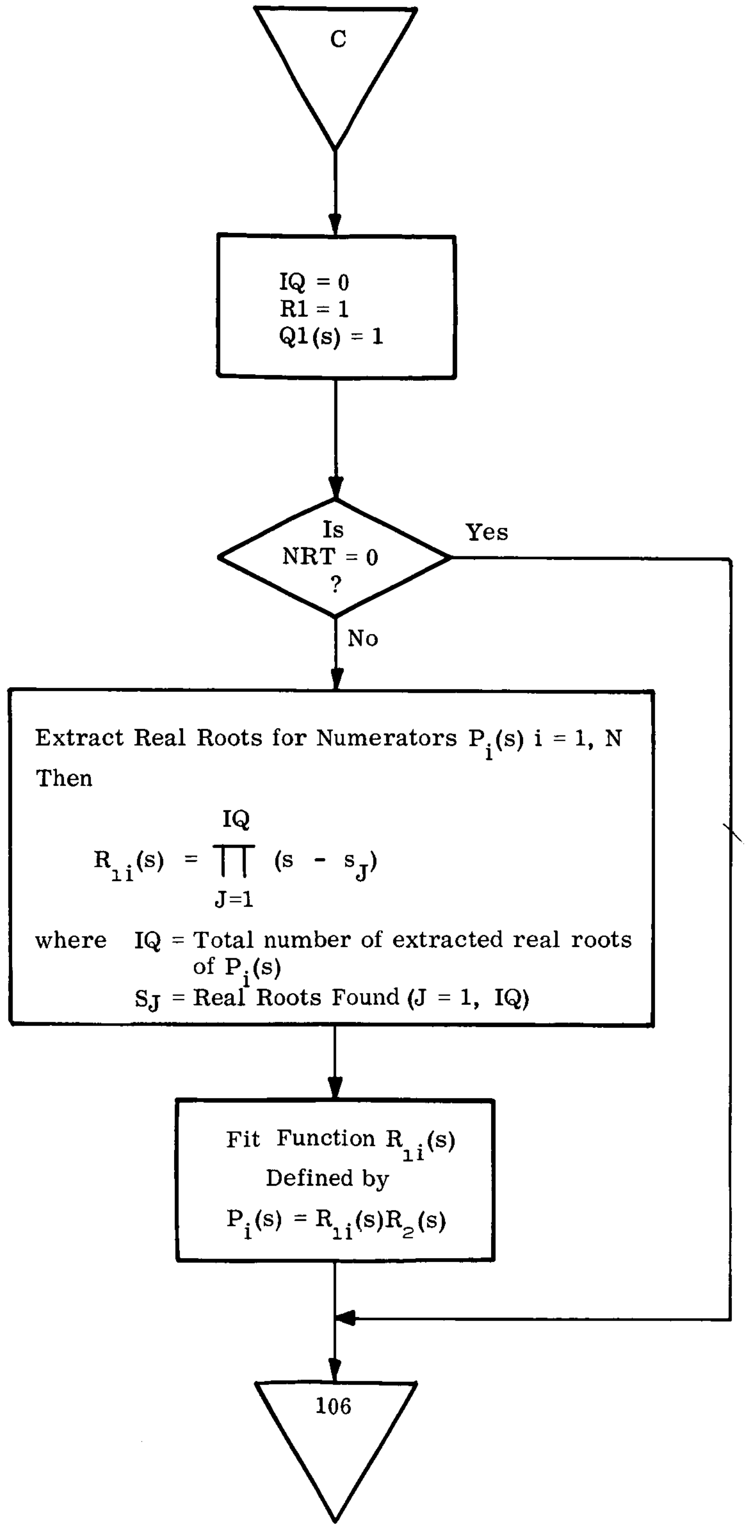



Figure C-4. Flow Chart 4 For Program CURFIT

# APPENDIX D

# CALCULATION OF SYSTEM TIME RESPONSE BY NUMBER SERIES FROM SUBSYSTEM OR COMPONENT TIME RESPONSES

## D.1 INTRODUCTION

In Reference 1 a method was described for finding the time response of a system directly from the time responses of its components by using number series. A program for reducing a sixth-order matrix by this method was described, but it had not become operational by the end of the contract period. In order to attempt to make use of this program rather than to devise a new one, the gyro problem was arranged for solution in three steps, as described in the following. The program continued to misbehave, and the results discussed in this report were obtained on a desk calculator and on the time-sharing computer.

## D.2 GENERAL

The derivation of the transfer functions describing the X-loop of the guidance IMU may be carried out in several ways. The following is an example of one method.

The X-loop to be analyzed is illustrated in quasi-block diagram form in Figure D-1. The circuitry associated with the loop is presented in equivalent simplified form in Figure D-2, with edge admittance factors, $y_i$.

Figure D-2 indicates the presence of a ground, or reference, node at $\epsilon_2$. Consequently, in order to keep the total matrix size as small as possible, the RCL network is first cut at node $\epsilon_A$ and $\epsilon_p$ and equivalent paths, $t(\epsilon_A, \epsilon_o)$ and $t(\epsilon_p, \epsilon_o)$, determined in the manner suggested in Reference 1. These two paths then become elements in the overall transition matrix of the system.

In the following, edge transmission factors may be represented as $t_{ij}$ or equivalently as $t(i, j)$, interchangeably.

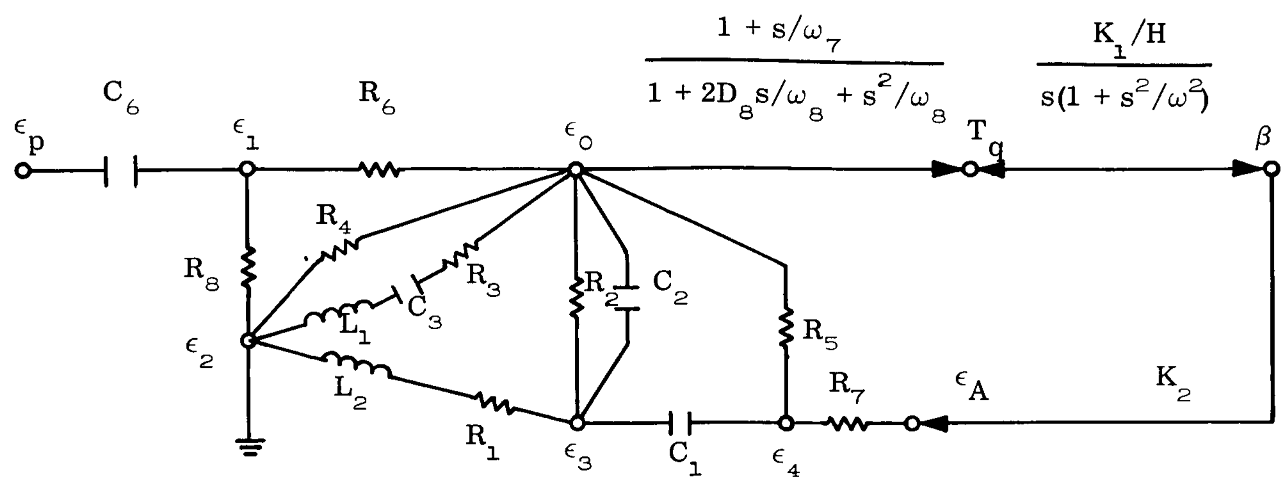



Figure D-1. X-Loop

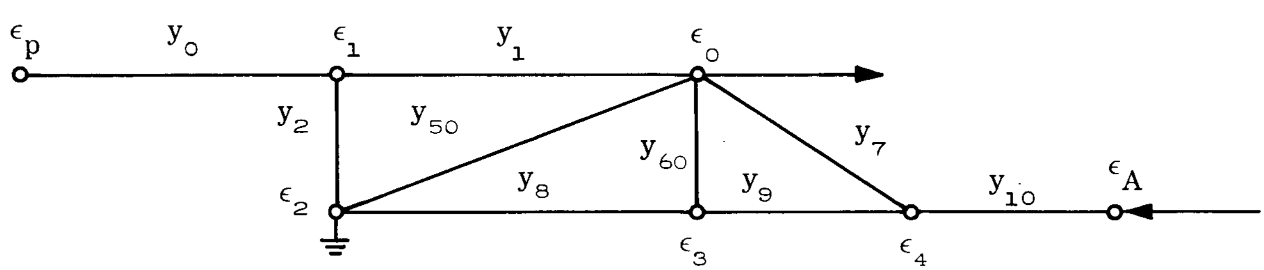



Figure D-2. Equivalent RCL Network

## D.3 EQUIVALENT PATHS $t(\epsilon_p, \epsilon_o)$ AND $t(\epsilon_A, \epsilon_o)$

The equivalent path, $t(\epsilon_p, \epsilon_o)$, from the input node $\epsilon_p$ to the output node $\epsilon_o$ is calculated from the flow graph:

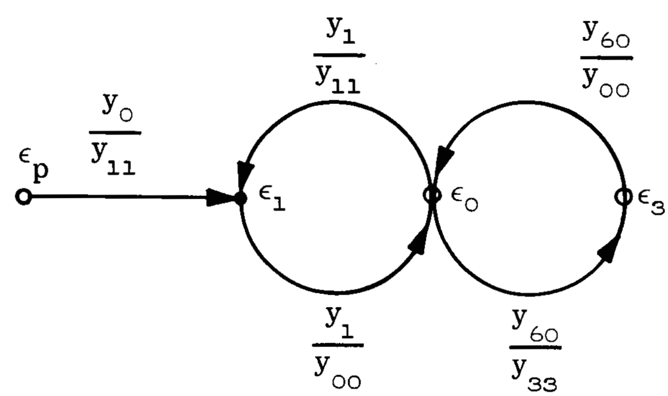



Figure D-3. Equivalent Path

where Figure D-3 is obtained directly from inspection of Figure D-2, and the $y_i$'s are admittance factors as subsequently defined. The network of Figure D-3 yields the transition matrix:

$$T(\epsilon_p, \epsilon_o) = \begin{array}{c|cccc} & \epsilon_p & \epsilon_1 & \epsilon_o & \epsilon_3 \\ \hline \epsilon_p & 1 & & & \\ \epsilon_1 & \frac{y_o}{y_{11}} & & \frac{y_1}{y_{11}} & \\ \epsilon_o & & \frac{y_1}{y_{00}} & & \frac{y_{60}}{y_{00}} \\ \epsilon_3 & & & \frac{y_{60}}{y_{33}} & \end{array} \qquad \text{(D-1)}$$

The equivalent path, $t(\epsilon_A, \epsilon_o)$, from the input node, $\epsilon_A$, to the output node, $\epsilon_o$, is obtained in like manner, as shown in Figure D-4.

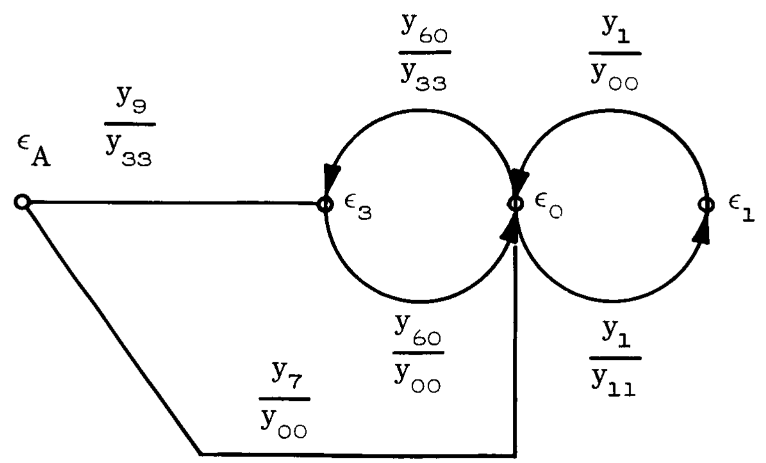



Figure D-4. Second Equivalent Path

Figure D-4 is obtained directly from inspection of Figure D-2. The network of Figure D-4 yields the transition matrix:

$$
T(\epsilon_A, \epsilon_0) = \begin{array}{c|cccc}
 & \epsilon_A & \epsilon_3 & \epsilon_1 & \epsilon_0 \\
\hline
\epsilon_A & 1 & & & \\
\epsilon_3 & \frac{y_9}{y_{33}} & & & \frac{y_{60}}{y_{33}} \\
\epsilon_1 & & & & \frac{y_1}{y_{11}} \\
\epsilon_0 & \frac{y_7}{y_{00}} & \frac{y_{60}}{y_{00}} & \frac{y_1}{y_{00}} &
\end{array} \qquad \text{(D-2)}
$$

The first two steps are to reduce the matrices (D-1) and (D-2) to $t(\epsilon_p, \epsilon_0)$ and $t(\epsilon_A, \epsilon_0)$.

## D.4 CLOSED LOOP SYSTEM

Having calculated the two equivalent paths, $t(\epsilon_p, \epsilon_0)$ and $t(\epsilon_A, \epsilon_0)$, through the networks, the overall circuit of Figure D-1 may be replaced with Figure D-5.

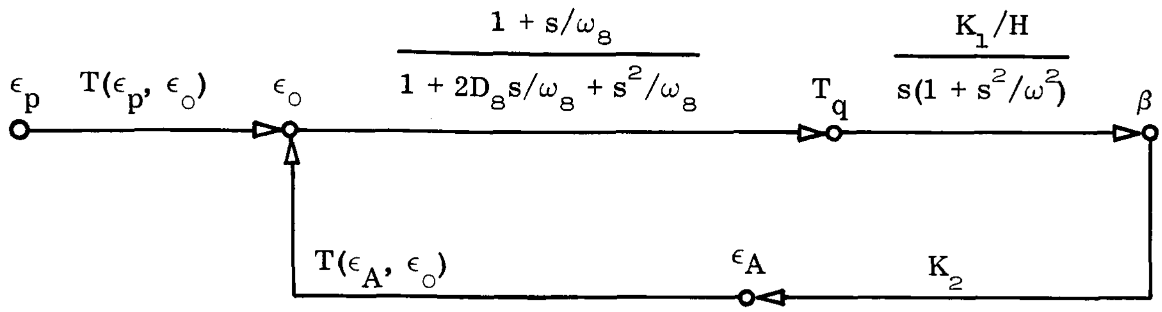



Figure D-5. Closed Loop Equivalent Path

The corresponding transition matrix, for the graph of Figure D-5, is:

$$T(\epsilon_p, \beta) = \begin{array}{c|ccccc} & \epsilon_p & \epsilon_o & T_q & \beta & \epsilon_A \\ \hline \epsilon_p & 1 & & & & \\ \epsilon_o & t(\epsilon_p, \epsilon_o) & & & & t(\epsilon_A, \epsilon_o) \\ T_q & & t(\epsilon_o, T_q) & & & \\ \beta & & & t(T_q, \beta) & & \\ \epsilon_A & & & & K_2 & \end{array} \qquad \text{(D-3)}$$

The equivalent path between nodes $\epsilon_p$ and $\beta$ may now be determined by reduction of $T(\epsilon_p, \beta)$ in the usual manner.

The numerical values to determine the $t_{ij}$ elements of the above three matrices are obtained from previously reported results, Reference 2. Conversion to the admittance factors used in this report may be made via the definitions provided in Table D-1, and in explicit form in Table D-2.

## D.5 MODIFIED COMPUTATIONAL SCHEME

Examination of the X-loop edge factors indicates that for the edges $y_0/y_{11}$, $y_{60}/y_{00}$, and $y_{60}/y_{33}$, and $y_9/y_{33}$, the degree of the numerator polynomial is equal to the degree

Table D-1

Definitions of Circuit Elements

| | |
|---|---|
| $y_0 = C_6 s$ | $y_5 = 1/R_2$ |
| $y_1 = 1/R_6$ | $y_6 = C_2 s$ |
| $y_2 = 1/R_8$ | $y_7 = 1/R_5$ |
| $y_3 = 1/R_4$ | $y_8 = 1/(R_1 + L_2 s)$ |
| $y_4 = (R_3 + L_1 s + 1/C_3 s)^{-1} = C_3 s/(1 + R_3 C_3 s + L_1 C_3 s^2)$ | |
| $y_9 = C_1 s$ | $y_{10} = 1/R_7$ |
| $y_{50} = y_3 + y_4 = \dfrac{1 + (R_3 C_3 + R_4 C_3)s + L_1 C_3 s^2}{R_4 + R_4 R_3 C_3 s + L_1 R_4 C_3 s^2}$ | |
| $y_{60} = y_5 + y_6 = \dfrac{1 + R_2 C_2 s}{R_2}$ | |
| $y_{11} = y_0 + y_1 + y_2 = \dfrac{R_6 + R_8 + R_8 R_6 C_6 s}{R_6 R_8}$ | |
| $y_{33} = y_8 + y_9 + y_{60} = \dfrac{(R_1 + R_2) + (R_2 R_1 C_1 + R_1 R_2 C_2 + L_2)s + (L_2 R_2 C_1 + L_2 R_2 C_2)s^2}{R_1 R_2 + R_2 L_2 s}$ | |
| $y_{44} = y_9 + y_7 + y_{10} = \dfrac{R_5 + R_7 + R_5 R_7 C_1 s}{R_5 R_7}$ | |
| $y_{00} = y_1 + y_{50} + y_{60} + y_7 = \dfrac{R_2 R_5 + R_2 R_6 + R_5 R_6 (1 + R_2 C_2 s)}{R_2 R_5 R_6} + \dfrac{1 + (R_3 C_3 + R_4 C_3)s + L_1 C_3 s^2}{R_4 + R_4 R_3 C_3 s + L_1 R_4 C_3 s^2}$ | |

Table D-2

X-Loop Numerical Values and Edge Factors

$y_o = 1 \times 10^{-6} s$

$y_1 = 6.25 \times 10^{-6}$

$y_2 = 1 \times 10^{-5}$

$y_3 = 2.631578 \times 10^{-5}$

$y_9 = 1 \times 10^{-6} s$

$y_5 = 1.960784 \times 10^{-5}$

$y_6 = 6.8 \times 10^{-7} s$

$y_7 = 5 \times 10^{-6}$

$y_8 = 2.127659 \times 10^{-4}$

$$y_{50} = \frac{1 + 1.201500s + 3.00 \times 10^{-4} s^2}{3.8 \times 10^4 + 2.337 \times 10^3 s + 1.14 \times 10^1 s^2}$$

$$y_{60} = (1 + 3.468 \times 10^{-2} s)/5.1 \times 10^4$$

$$y_{11} = (2.6 \times 10^5 + 1.6 \times 10^4 s)/1.6 \times 10^{+10}$$

$$y_{33} = (5.57 \times 10^4 + 4.02696 \times 10^2 s)/2.397 \times 10^8$$

$$y_{00} = \frac{2.17259792 + 1.2994988s + 22.409394 \times 10^{-4} s^2 + 7.752 \times 10^{-6} s^3}{3.8 \times 10^4 + 2.337 \times 10^3 s + 11.4 s^2}$$

$$y_4 = 3 \times 10^{-5} s/(1 + 6.15 \times 10^{-2} s + 3 \times 10^{-4} s^2)$$

$$\frac{y_o}{y_{11}} = \frac{6.153846 \times 10^{-2} s}{1 + 6.153846 \times 10^{-2} s} = \frac{s}{s + 16.25}$$

$$\frac{y_1}{y_{11}} = \frac{38.4615375 \times 10^{-2}}{1 + 6.153846 \times 10^{-2} s} = \frac{6.25}{s + 16.25}$$

$$\frac{y_1}{y_{00}} = \frac{23.75 \times 10^{-2} + 1.460625 \times 10^{-2} s + 7.125 \times 10^{-5} s^2}{2.17259792 + 1.2994988s + 22.4093941 \times 10^{-4} s^2 + 7.752 \times 10^{-6} s^3}$$

$$\frac{y_{60}}{y_{00}} = \frac{0.74509803 + 7.1663529 \times 10^{-2} s + 1.81268941 \times 10^{-3} s^2 + 7.752 \times 10^{-6} s^3}{2.17259792 + 1.2994988s + 22.4093941 \times 10^{-4} s^2 + 7.752 \times 10^{-6} s^3}$$

$$\frac{y_{60}}{y_{33}} = \frac{1 + 3.468 \times 10^{-2} s}{11.85106329 + 8.568 \times 10^{-2} s}$$

$$\frac{y_9}{y_{33}} = \frac{4.303411 \times 10^{-3} s}{1 + 7.229731 \times 10^{-3} s}$$

$$\frac{y_7}{y_{00}} = \frac{0.190 + 0.011685s + 57 \times 10^{-6} s^2}{2.17259792 + 1.2994988s + 22.40939441 \times 10^{-4} s^2 + 7.752 \times 10^{-6} s^3}$$

$$t_{\epsilon_o, T_q} = \frac{1 + s/5499}{1 + \frac{0.68}{1759} s + \frac{s^2}{1759^2}} = \frac{1 + 1.8185125 \times 10^{-4} s}{1 + 3.8658329 \times 10^{-4} s + 3.2319774 \times 10^{-7} s^2}$$

$$t_{T_q, \beta} = \frac{73439.7754}{s(s^2 + 1521)} = \frac{73439.7754}{s(s^2 + 39^2)}$$

$$t_{\beta, \epsilon_A} = 572.958$$

of the denominator polynomial. Consequently, an impulse must be present in the number series equivalent to these edges.

These impulses may be removed by dividing the denominator into the numerator and injecting an extra node into each parallel edge created by the division. This method gives matrices (D-4), (D-5), and (D-6) to be reduced instead of (D-1), (D-2), and (D-3).

The modified $T(\epsilon_p, \epsilon_o)$ matrix is $T'(\epsilon_p, \epsilon_o)$

$$T'(\epsilon_p, \epsilon_o) = \begin{array}{c|ccccccc}
 & \epsilon_p & \epsilon'_p & \epsilon_1 & \epsilon_o & \epsilon'_o & \epsilon_3 & \epsilon'_3 \\
\hline
\epsilon_p & 1 & & & & & & \\
\epsilon'_p & 1 & & & & & & \\
\epsilon_1 & \dfrac{y'_o}{y_{11}} & 1 & & \dfrac{y_1}{y_{11}} & & & \\
\epsilon_o & & & \dfrac{y_1}{y_{oo}} & & & \dfrac{y'_{60}}{y_{oo}} & 1 \\
\epsilon'_o & & & & 0.4047619 & & & \\
\epsilon_3 & & & & \dfrac{y'_{60}}{y_{33}} & 1 & & \\
\epsilon'_3 & & & & & & 1 &
\end{array} \qquad \text{(D-4)}$$

Reduction of the $T'(\epsilon_p, \epsilon_o)$ matrix, in the usual manner, yields the equivalent path as the sum of two terms

$$t(\epsilon_p, \epsilon_o) = 1 + t'(\epsilon_p, \epsilon_o) \qquad \text{(D-5)}$$

The modified $T(\epsilon_A, \epsilon_o)$ matrix, $T'(\epsilon_A, \epsilon_o)$, is:

$$
T'(\epsilon_A, \epsilon_o) = \begin{array}{c|ccccccc}
 & \epsilon_A & \epsilon'_A & \epsilon_3 & \epsilon_o & \epsilon_1 & \epsilon'_o & \epsilon'_3 \\
\hline
\epsilon_A & 1 & & & & & & \\
\epsilon'_A & 0.59523805 & & & & & & \\
\epsilon_3 & \dfrac{y'_9}{y_{33}} & 1 & & \dfrac{y'_{60}}{y_{33}} & & 1 & \\
\epsilon_o & \dfrac{y_7}{y_{oo}} & & \dfrac{y'_{60}}{y_{oo}} & & \dfrac{y_1}{y_{oo}} & & 1 \\
\epsilon_1 & & & & \dfrac{y_1}{y_{11}} & & & \\
\epsilon'_o & & & & 1 & & & \\
\epsilon'_3 & & & 1 & & & &
\end{array} \qquad \text{(D-6)}
$$

The transition matrix becomes:

$$
T(\epsilon_p, \beta) = \begin{array}{c|cccccc}
 & \epsilon_p & \epsilon_o & T_q & \beta & \epsilon_A & \epsilon'_A \\
\hline
\epsilon_p & 1 & & & & & \\
\epsilon_o & t(\epsilon_p, \epsilon_o) & & & & t'(\epsilon_A, \epsilon_o) & 1 \\
T_q & & t(\epsilon_o, T_q) & & & & \\
\beta & & & t(T_q, \beta) & & & \\
\epsilon_A & & & & K_2 & & \\
\epsilon'_A & & & & & 1 &
\end{array} \qquad \text{(D-7)}
$$

## D.6 NUMBER SERIES DIVISION

The multiplication of number series provides little difficulty from an accuracy standpoint. It is equivalent to the numerical evaluation of the Convolution Integral; however, the situation with respect to division is sensitive to the method of defining the number series for the denominator term.

The following method has yielded satisfactory results on preliminary examples. The basic problem is the removal of self loops obtained in the current matrix reduction scheme.

Consider the two-node problem illustrated in Figure D-6,

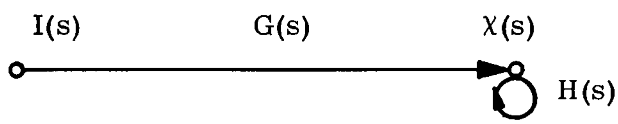



Figure D-6. Equivalent Path for Division

where I(s) is the input node and $\chi(s)$ the output node containing the self loops H(s). H(s) and G(s) are arbitrary transfer functions. Thus:

$$\chi(s) = \frac{G(s)}{1 - H(s)} \tag{D-8}$$

or

$$G(s) = \left[1 - H(s)\right] \chi(s) \tag{D-9}$$

Whence:

$$G(t) = \chi(t) - \int_0^t H(t - \tau)\chi(\tau)d\tau \tag{D-10}$$

which in number series form becomes:

$$\hat{g} = \hat{\chi} - \hat{h}\hat{\chi} \tag{D-11}$$

Equation D-10 is a Volterra Integral Equation to solve for $\chi(t)$. One numerical method for obtaining a solution is to express $\hat{\chi}$ in literal form, perform the number series multiplication indicated in Equation D-11, then evaluate the unknown parameters in (D-11). Using the Trapezoidal Rule for multiplication gives:

$$g_0, g_1, \ldots, g_n = \chi_0, \quad \left[\chi_1 - \left(\frac{h_0\chi_1}{2} + \frac{h_1\chi_0}{2}\right)\Delta t\right], \ldots . \quad \left[\chi_n - \left(\frac{h_0\chi_n}{2} + h_1\chi_{n-1} + \ldots + \frac{h_n\chi_0}{2}\right)\Delta t\right] \tag{D-12}$$

Equating corresponding terms and solving:

$$\chi_0 = g_0$$

$$\chi_1 = \frac{g_1 + h_1\chi_0\Delta t/2}{1 - h_0\Delta t/2}$$

$$\vdots$$

$$\chi_n = \frac{g_n + (h_1\chi_{n-1} + h_2\chi_{n-2} + \ldots h_n\chi_0/2)\Delta t}{1 - h_0\Delta t/2}$$

On simple matrix reduction problems similar to the gyro problem, this method of division has given more accurate results than some others described in published literature. This new division algorithm was substituted in the number series program to replace the older one.

## REFERENCES


1. Analytical Study of Launch Vehicle Component Level Simulation, December 1965.

2. Simulation of IMU; Part I - Platform and PIGA - 66 SIMAR 98/8, February 1966.

3. Investigation of Requirements for Multiple Engine Propulsion System Simulation, March 17, 1966 (Prepared by W. P. Crownover, E. W. Reinhardt, E. D. Fisher).